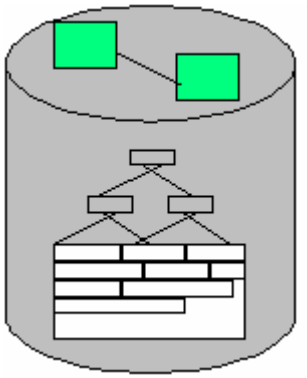
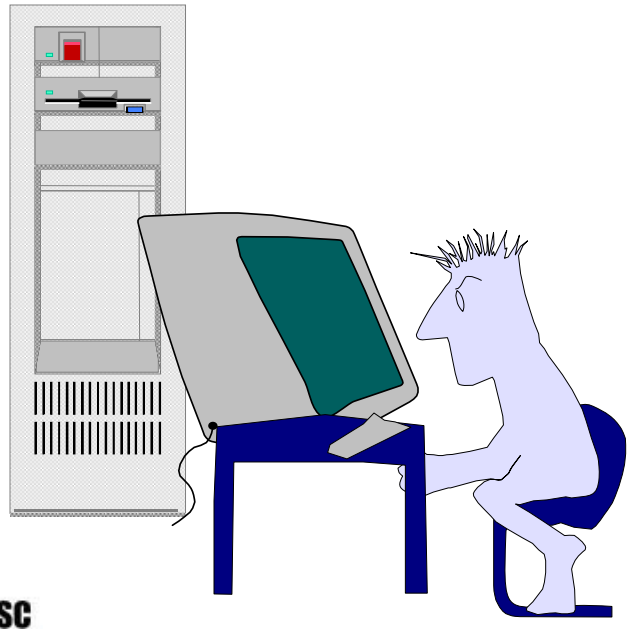


Database Design Process

Duration: 12hrs



A mini-world example

- A Company is organised in to departments. Each department has a number and an employee who manages the department. We keep track of the start date when that employee started managing the department. A department may have several locations.
- A department controls a number of projects. Each of which has a name, a number and a single location.

A mini-world example

- We store each employee's name, national Id number, address, salary, birth date and sex. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled, by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.

A mini-world example

- We keep track of the dependants of each employee for insurance purposes. We keep each dependant's name, sex, birth date and relationship to the employee.

Such information is gathered from the mini-world to perform *Phase 1* of database design process. i.e. *Requirements Collection and Analysis Phase*

Conceptual Design

All the requirements collected at *Phase 1* are analysed to create a *Conceptual Schema*.

This process is called the *Conceptual Design*.

We identify the *entities*, their *attributes*, r*elationships* and *constraints* (business rules). The conceptual schema is used as a reference to ensure that all user's data requirements are met and the requirements do not include any conflicts.

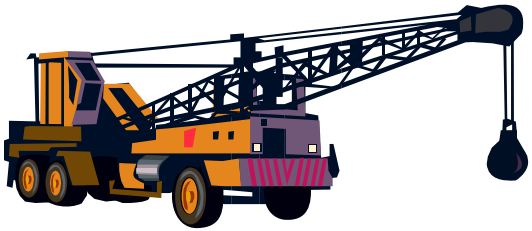
Conceptual Design

Entities

Department

Employee

Project

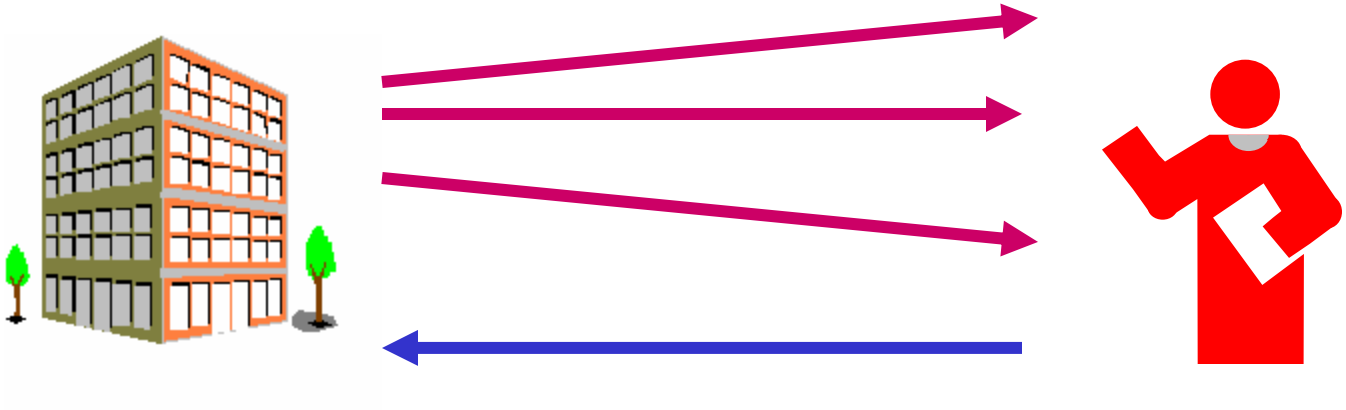


Dependent

Conceptual Design

Relationships

A **Department** has Many **Employees**



An **Employee** works for A **Department**

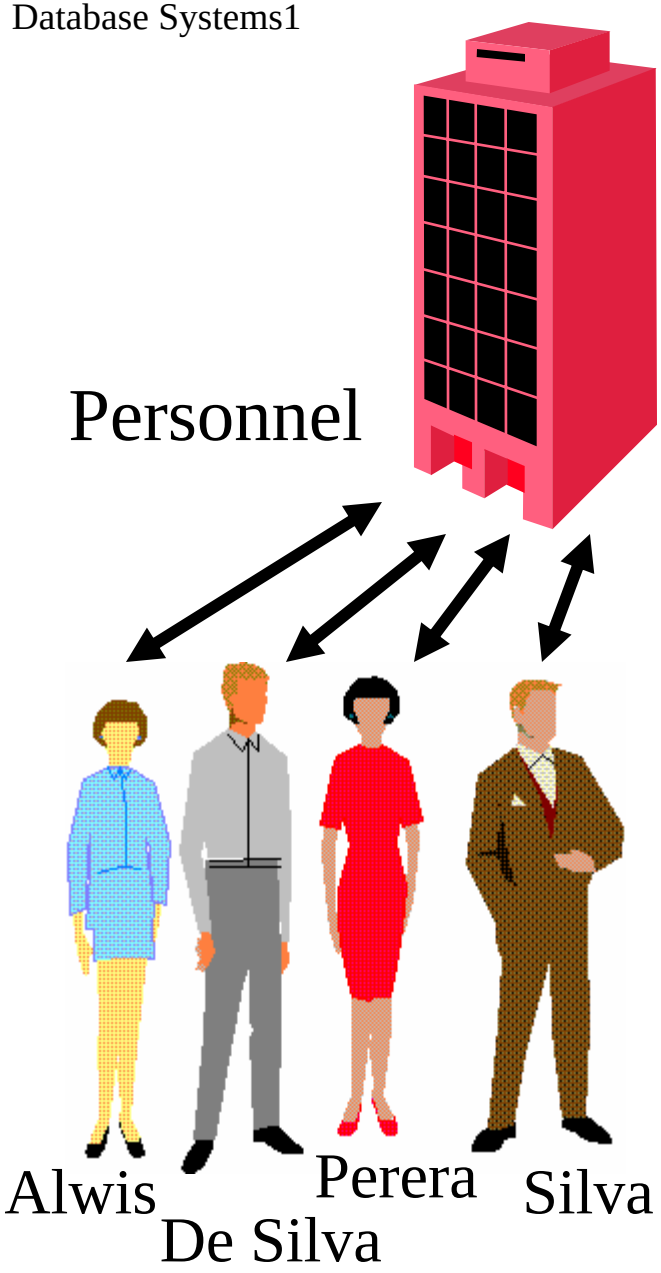
one to many relationship

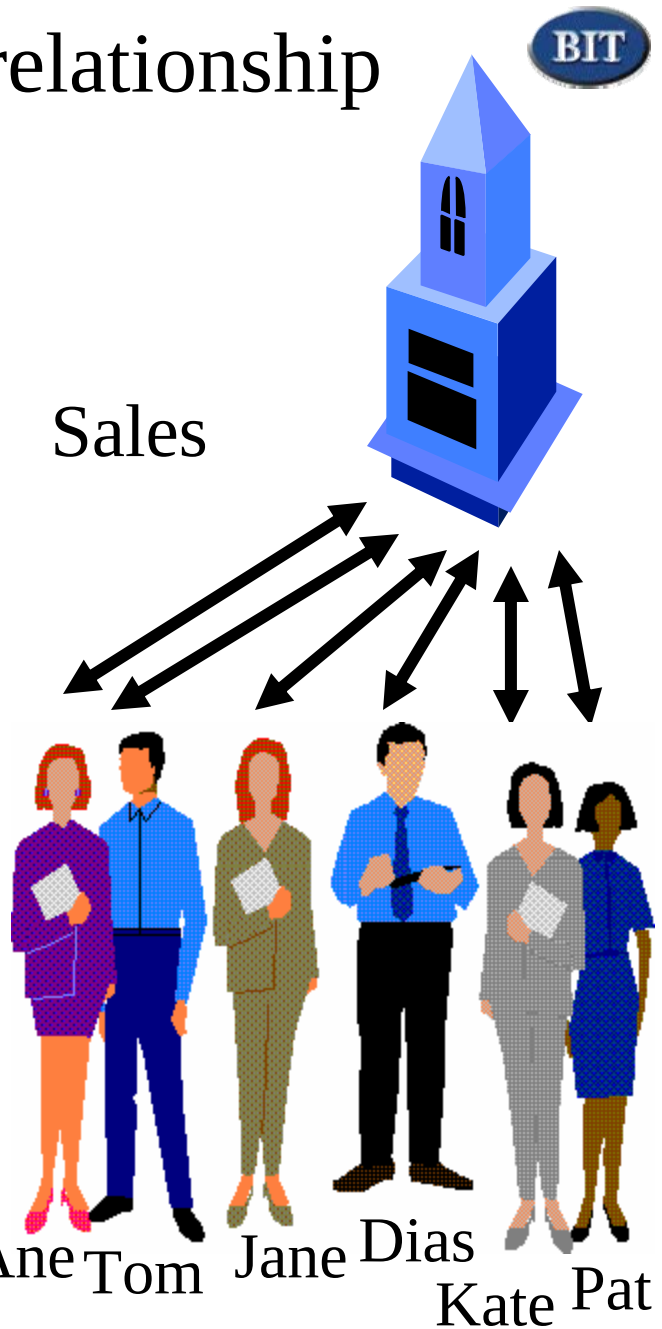
Department

Personnel

Sales

Employee





Relationships

- Relationship Type
  - A meaningful association between association between (or among) entity types
- Relationship Instances
  - An association between (or among) entity instances, where each relationship instance includes exactly one entity from each participating entity type

  e.g. De Silva works for Personnel Department

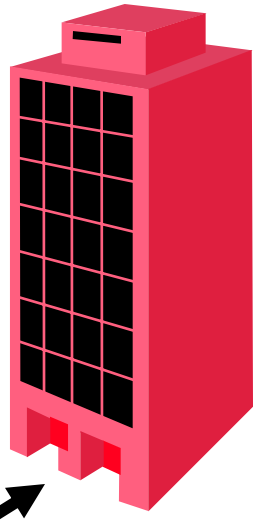
Conceptual Design

Relationships

A **Department** has A Manager **(Employee)**

An **Employee** manage A **Department**

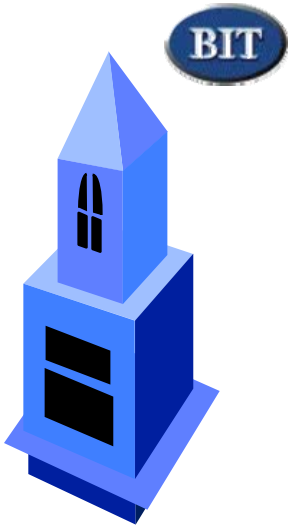
one to one relationship

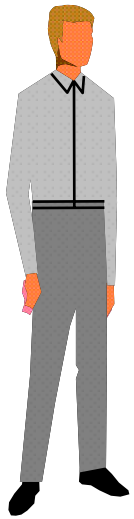


Department

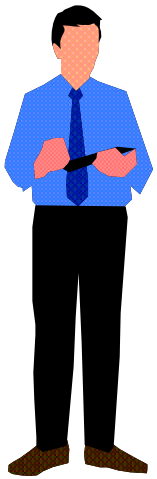
Personnel

Sales





Manager
(Employee)



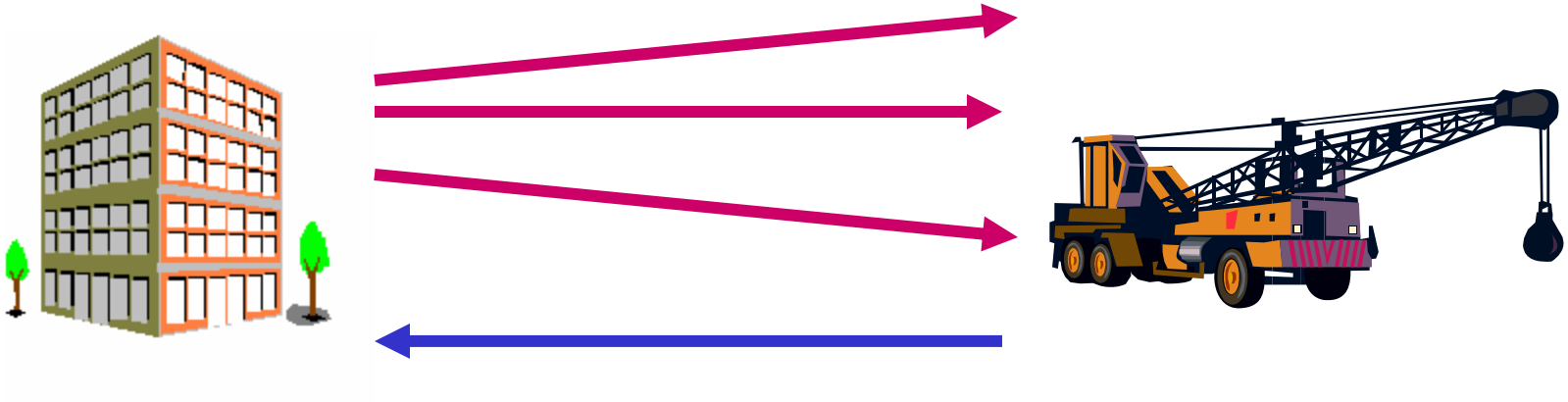
De Silva

Dias

Conceptual Design

Relationships

A **Department** controls Many **Projects**

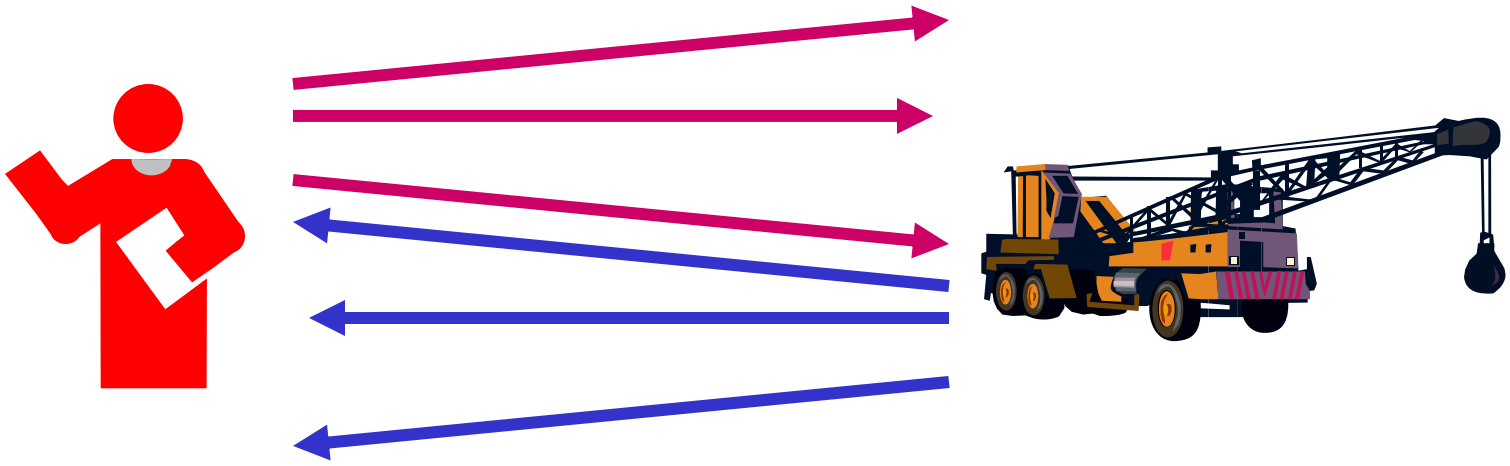


A **Project** controlled by A **Department**

Conceptual Design

Relationships

An **Employee** works on Many **Projects**

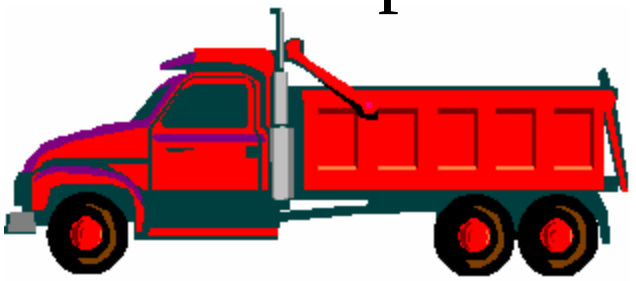


A **Project** has Many **Employees**

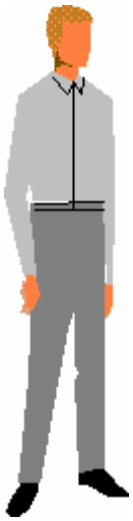
many to many relationship

Construction

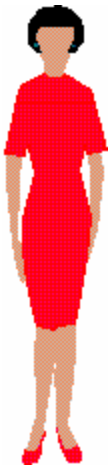
Project



Delivery



De Silva



Perera

Employee

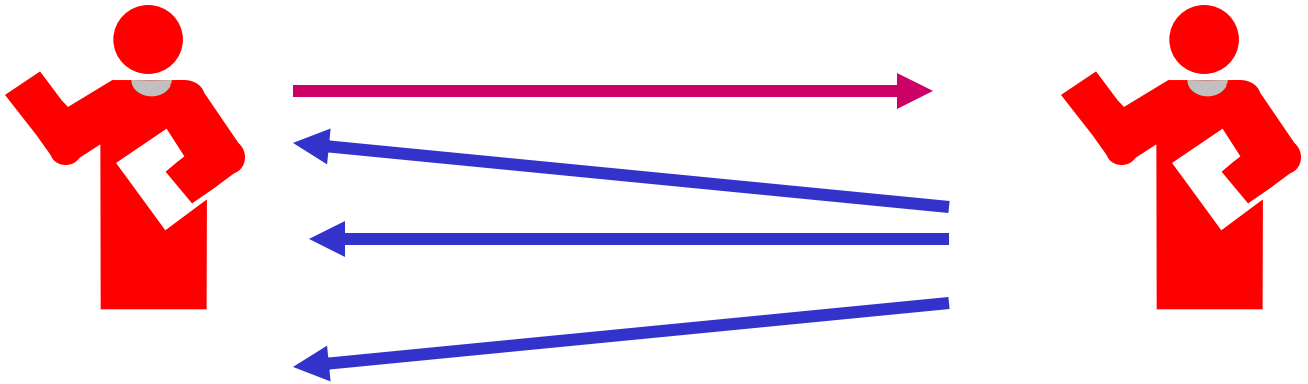
Dias

1 relationship type
5 relationship instances

Conceptual Design

Relationships

An **Employee** supervised by An **Employee**

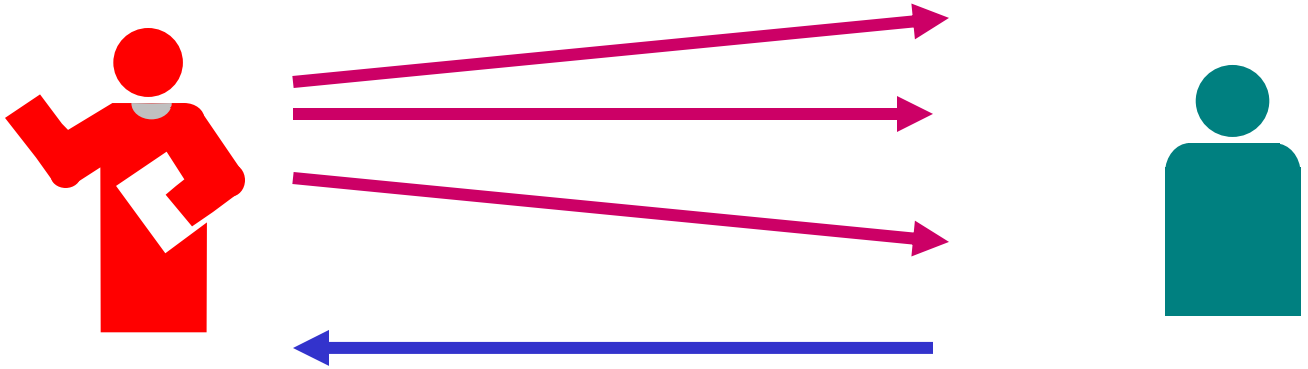


An **Employee** supervise Many **Employees**

Conceptual Design

Relationships

An **Employee** has Many **Dependants**

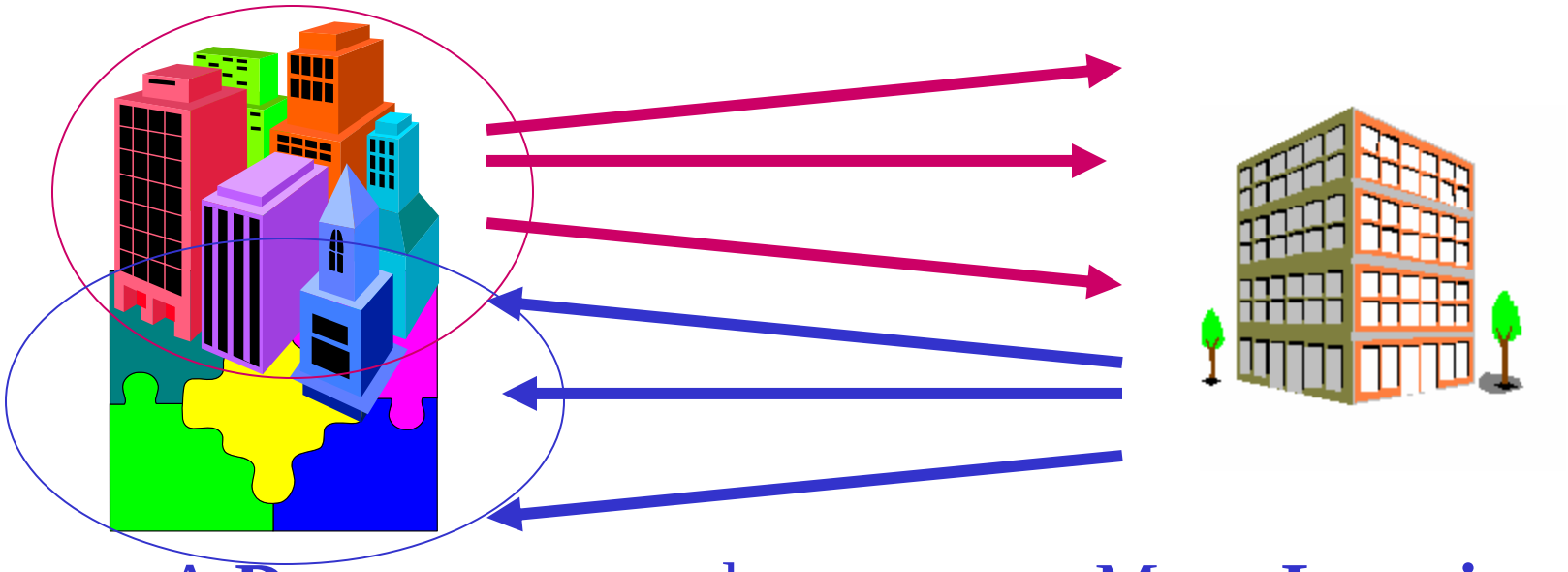


A **Dependant** belongs to An **Employee**

Conceptual Design

Entities / Relationships??

A **Company** has Many **Departments**

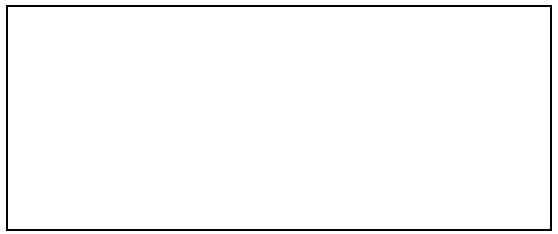


A **Department** has Many **Locations**

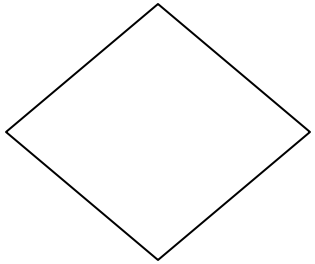
Conceptual Design

Notations

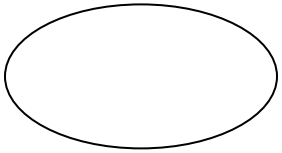
Entity



Relationship



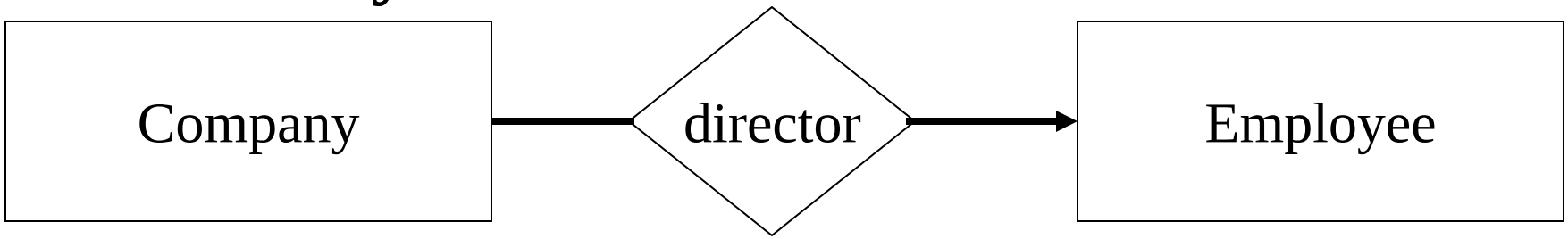
Attribute



Relationship Types

One to One

One to Many



Many to Many

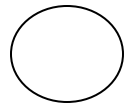
Cardinality Constraints

- Specifies the number of instances of one entity that can (or must) be associated with each instance of another entity
- Minimum Cardinality
  - The minimum number of instances of one entity that may be associated with each instance of another entity

  e.g. the minimum dependants for an employee is zero

Cardinality Constraints

- Optional Participation
  - when the number of participants in the relationship is zero



- Mandatory Participation
  - when the number of participants in the relationship is one
- Maximum Cardinality
  - The maximum number of instances of one entity that may be associated with a single occurrence of another entity

e.g. an Employee can have insurance policies for at most two dependants (0:2)

Existence Conditions

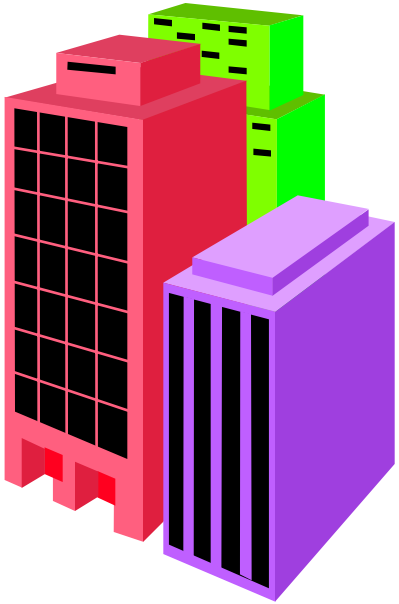
One to One

One to Many

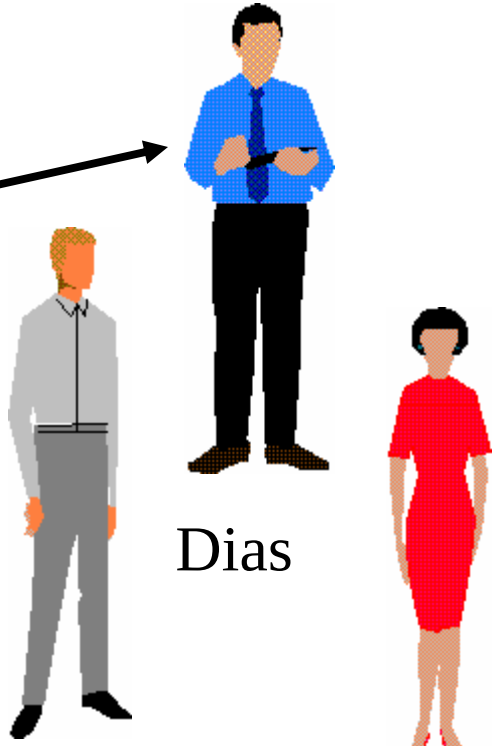
Many to Many

Existence Conditions

One to One



ABC Ltd.



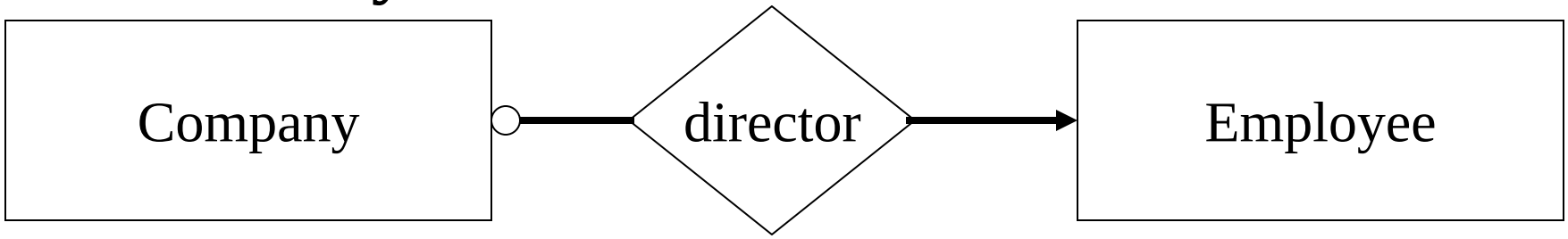
Dias

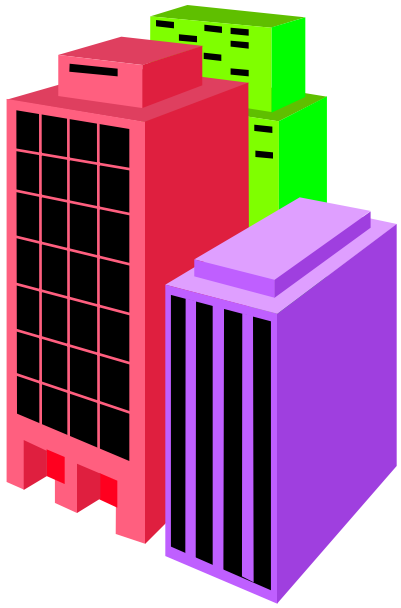
De Silva

Perera

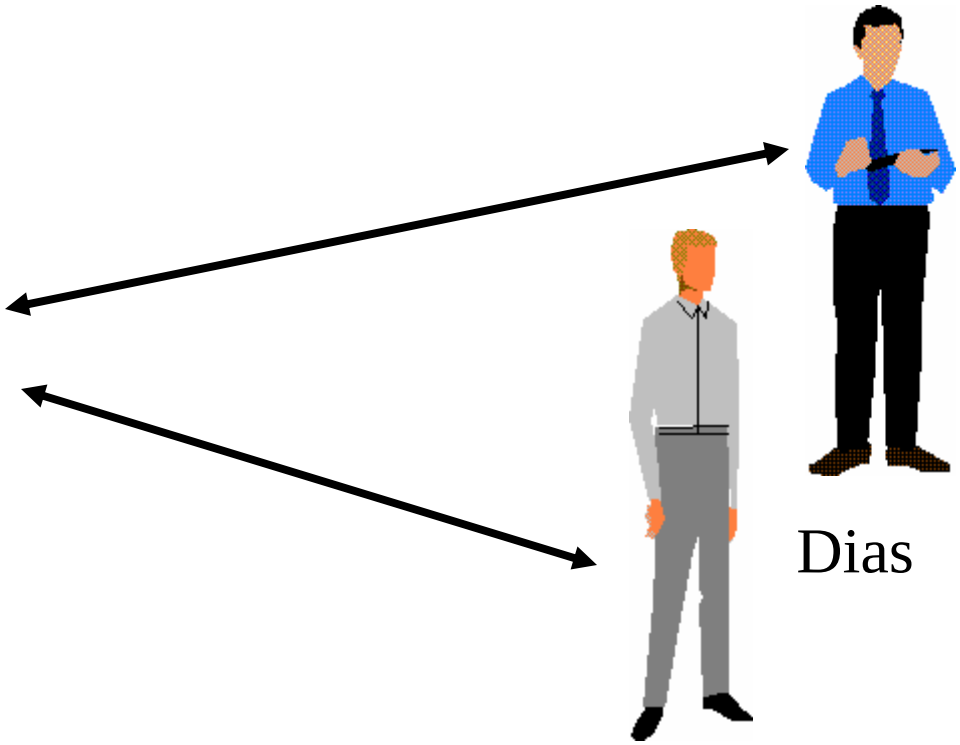
Existence Conditions

One to Many





ABC Ltd.



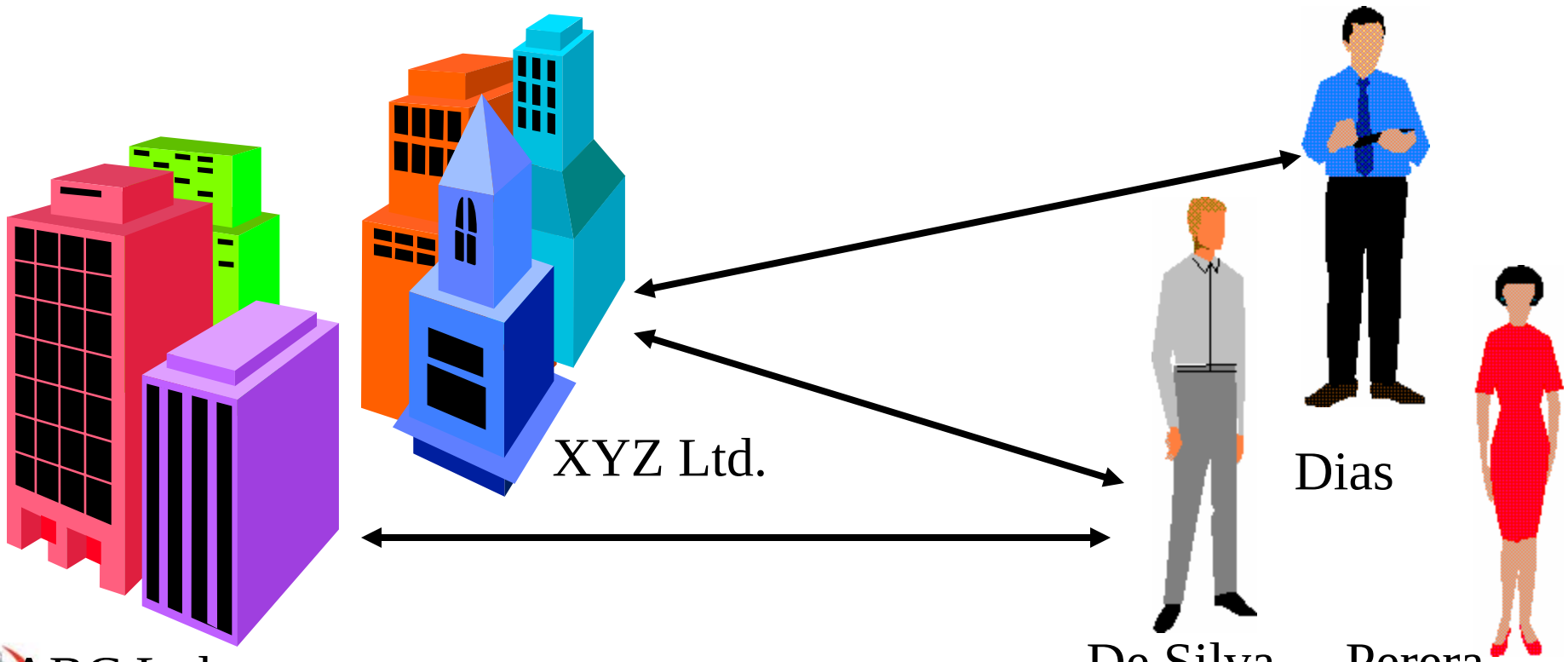
Dias

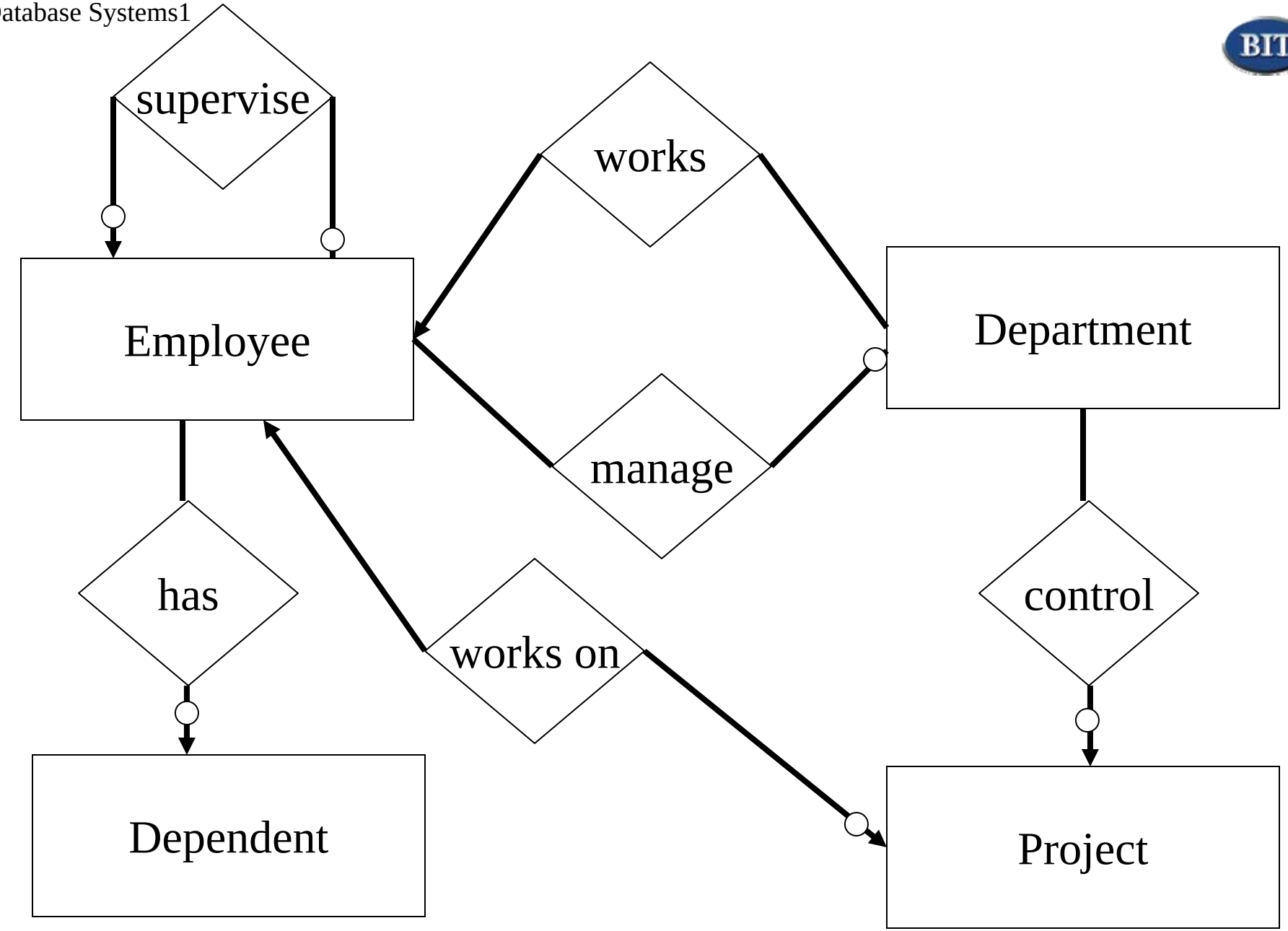
De Silva

Perera

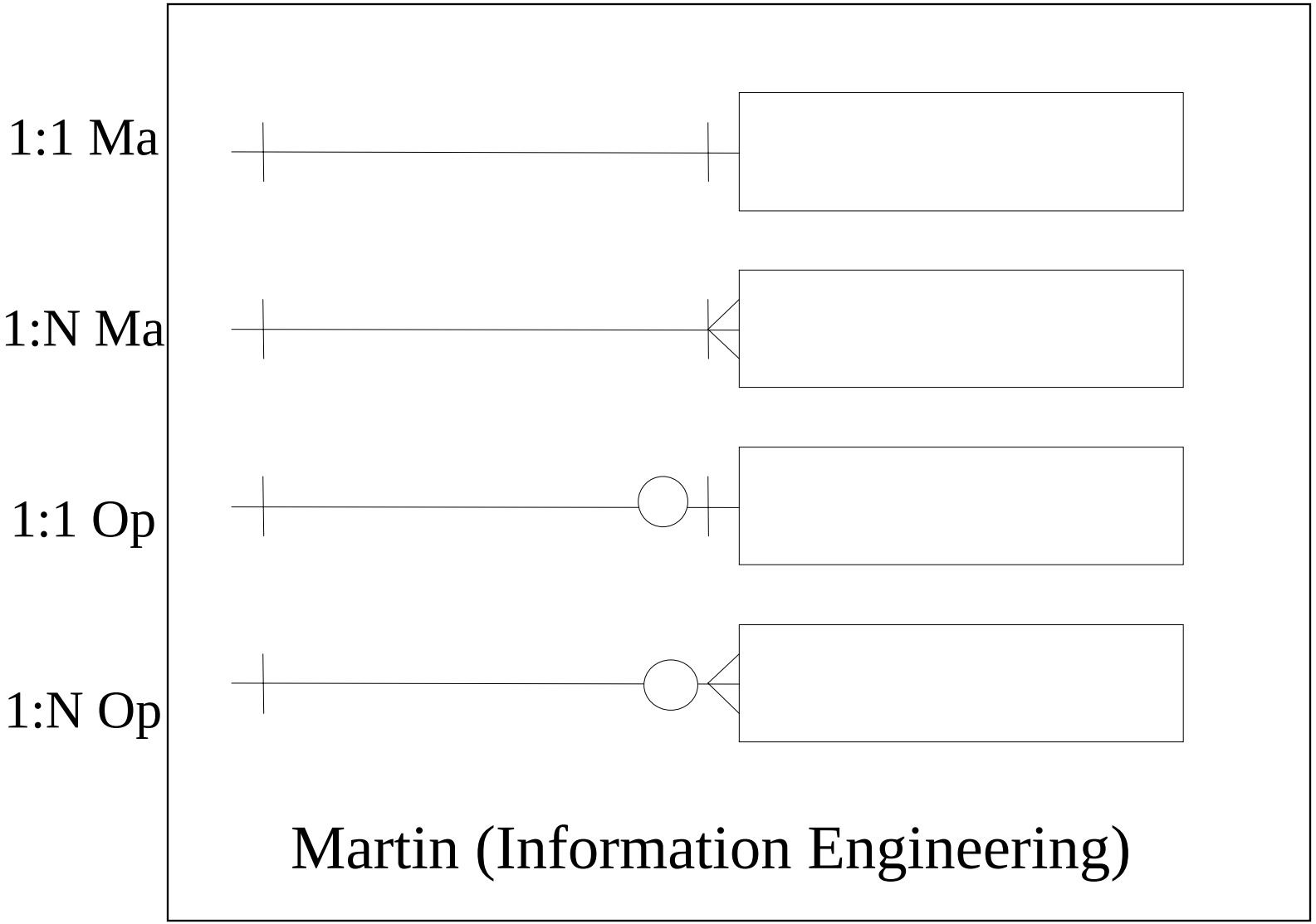
Existence Conditions

Many to Many

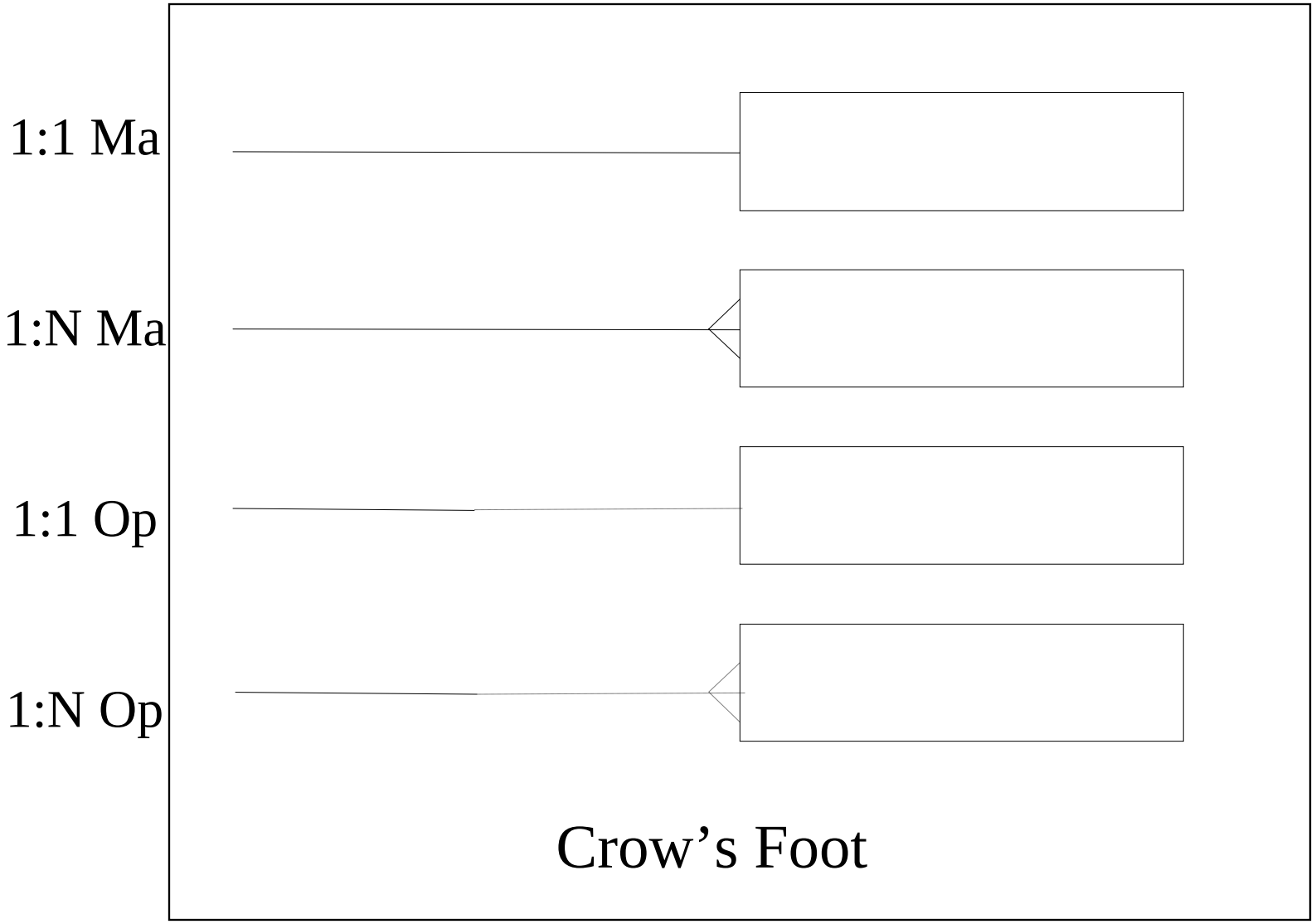



supervise
works
Employee
Department
manage
has
works on
control
Dependent
Project

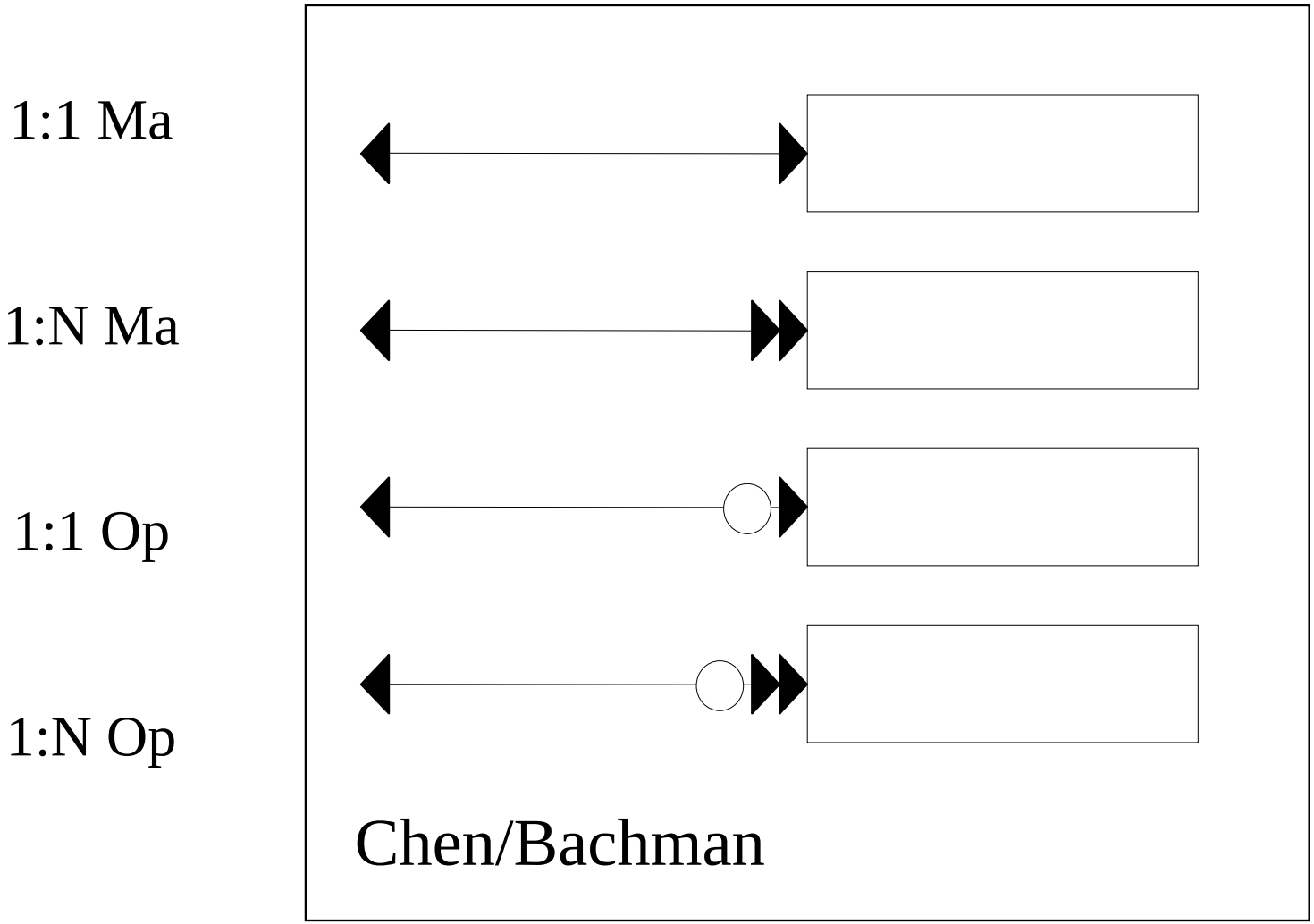
E-R Modelling



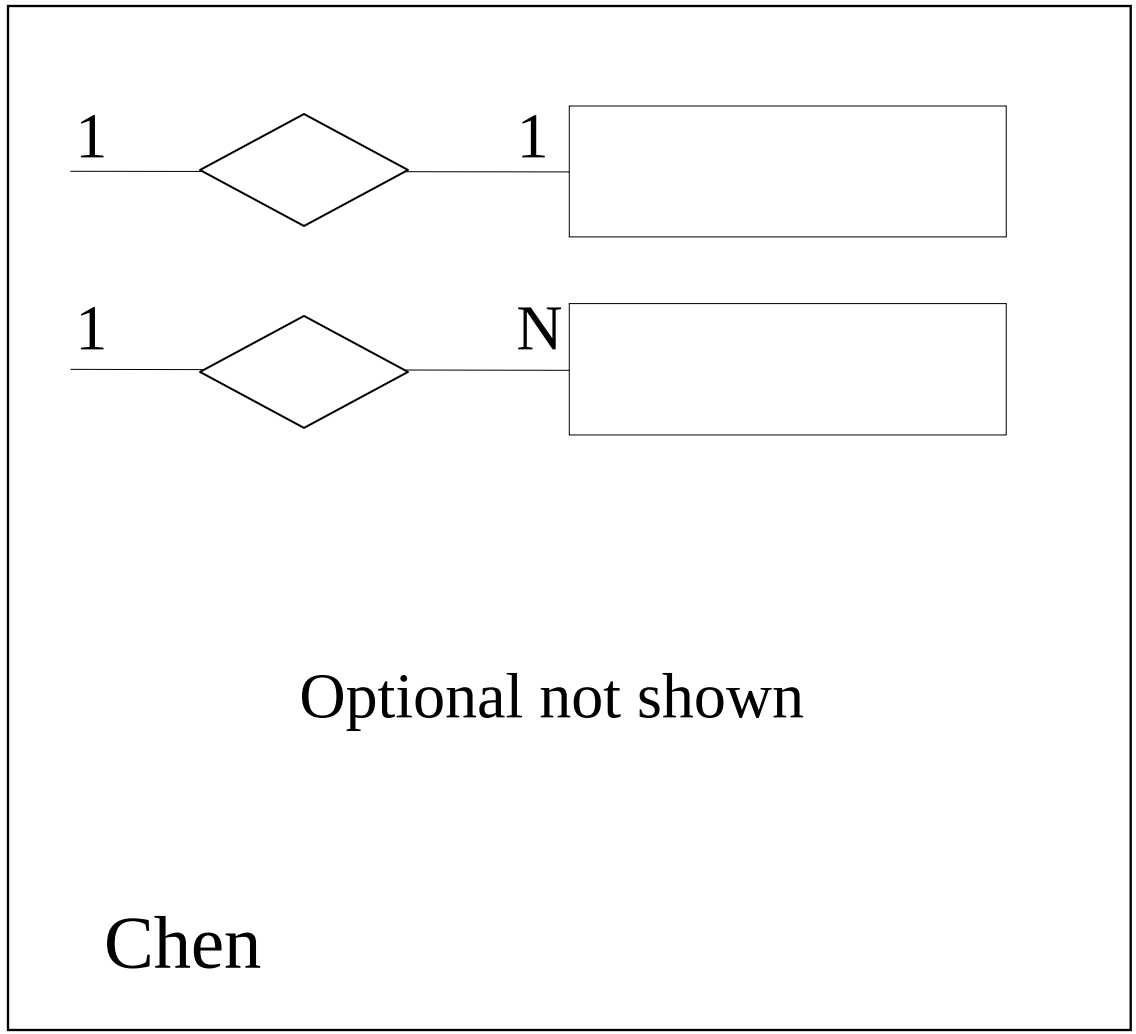
E-R Modelling



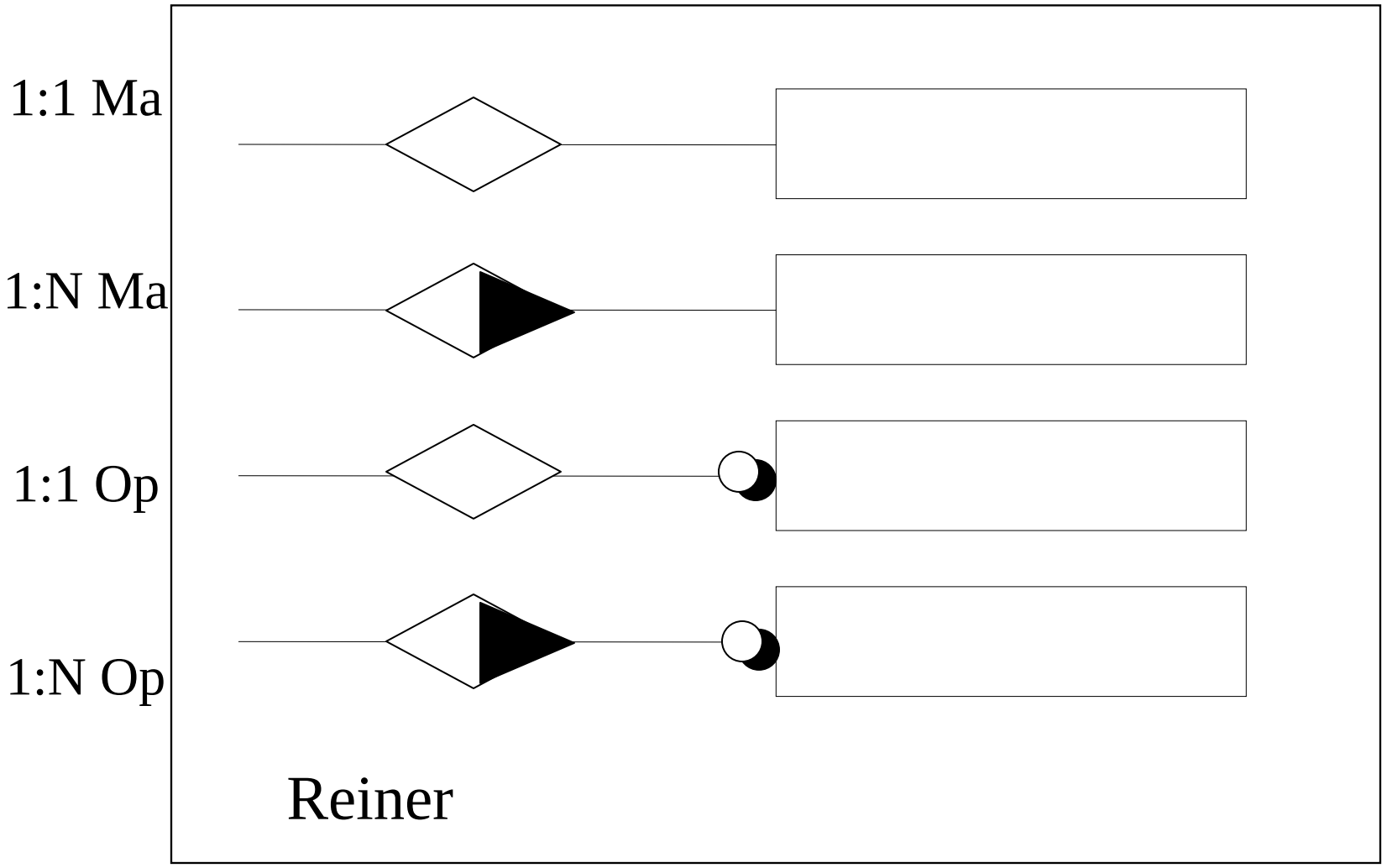
E-R Modelling (Alternate Notations)



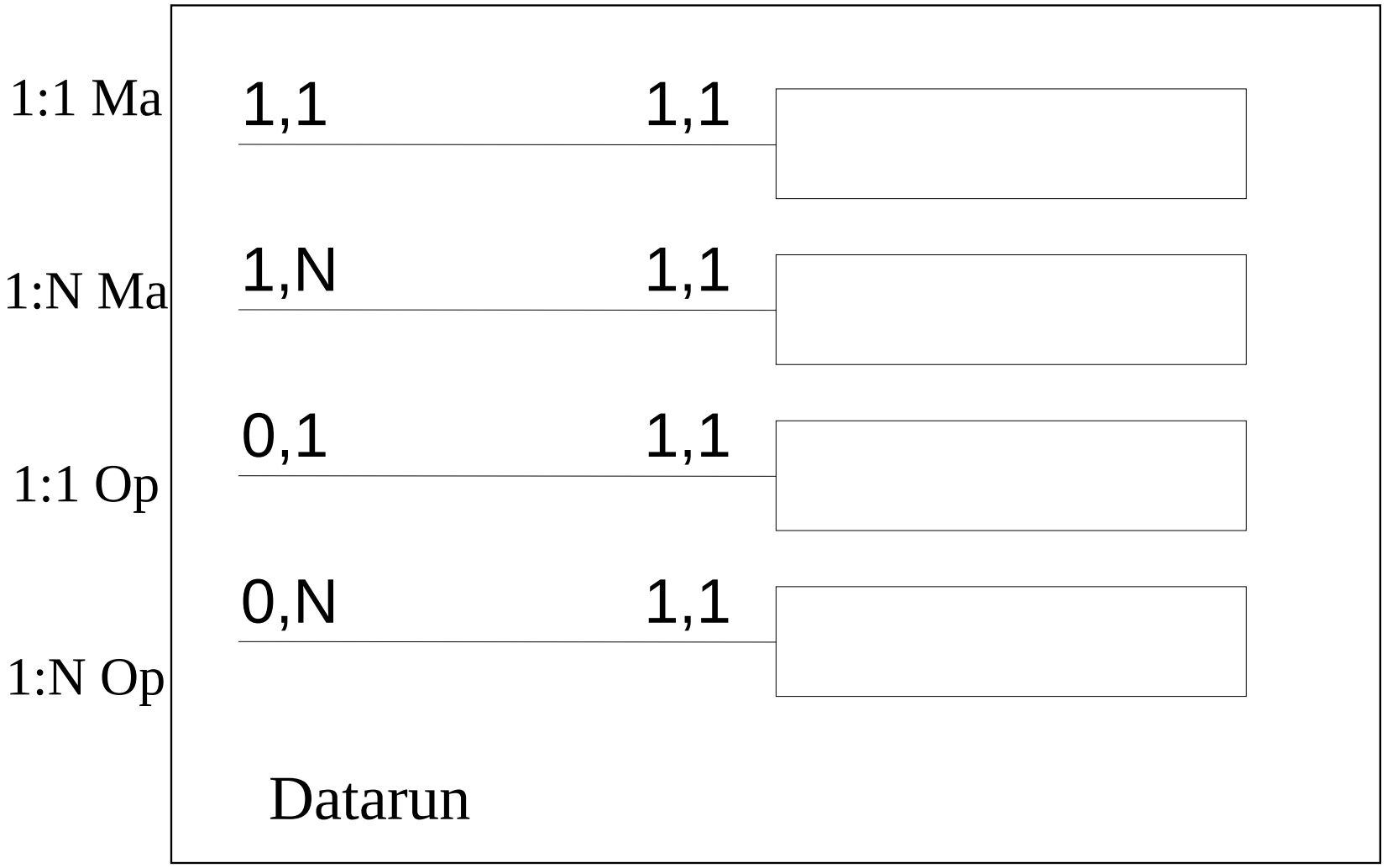
E-R Modelling (Alternate Notations)



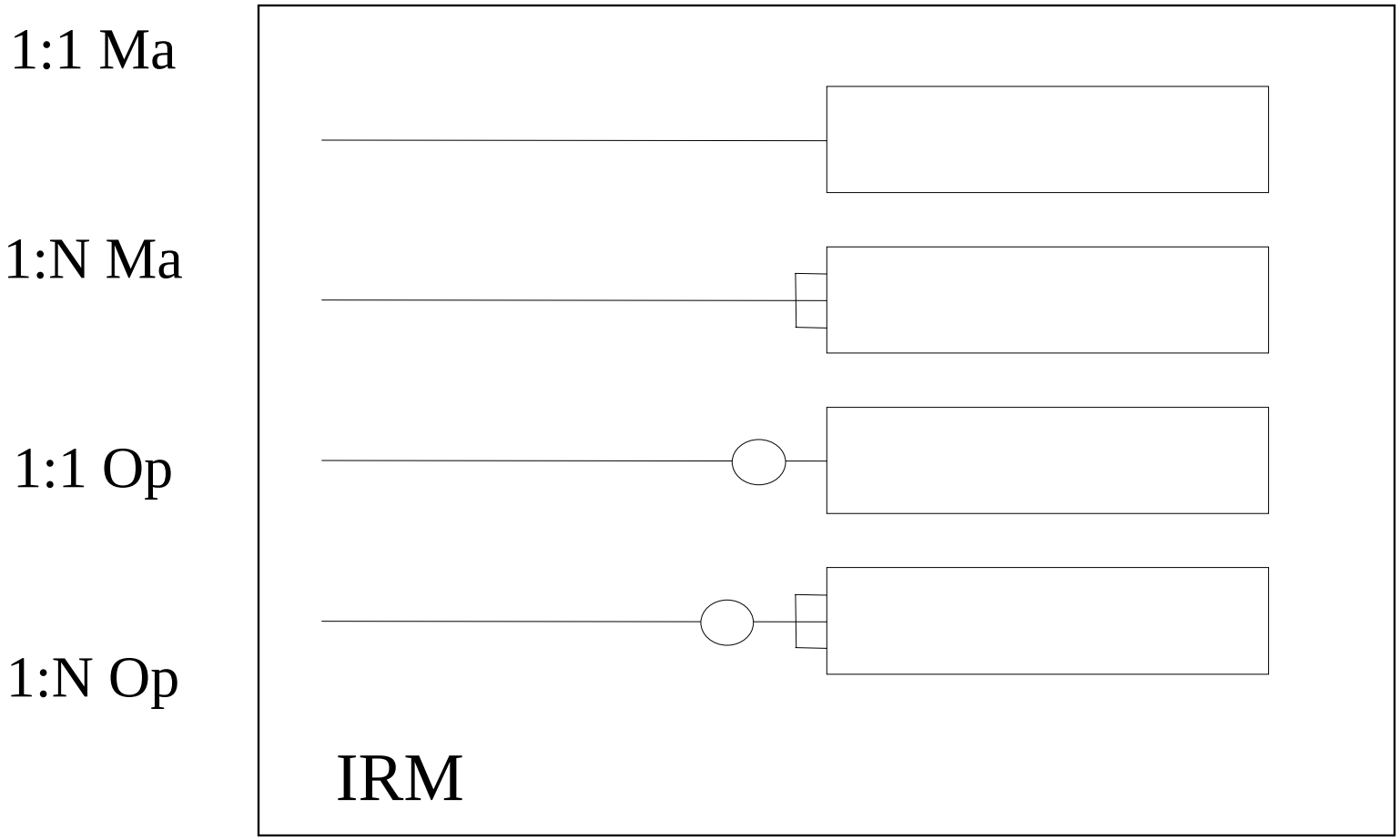
E-R Modelling (Alternate Notations)



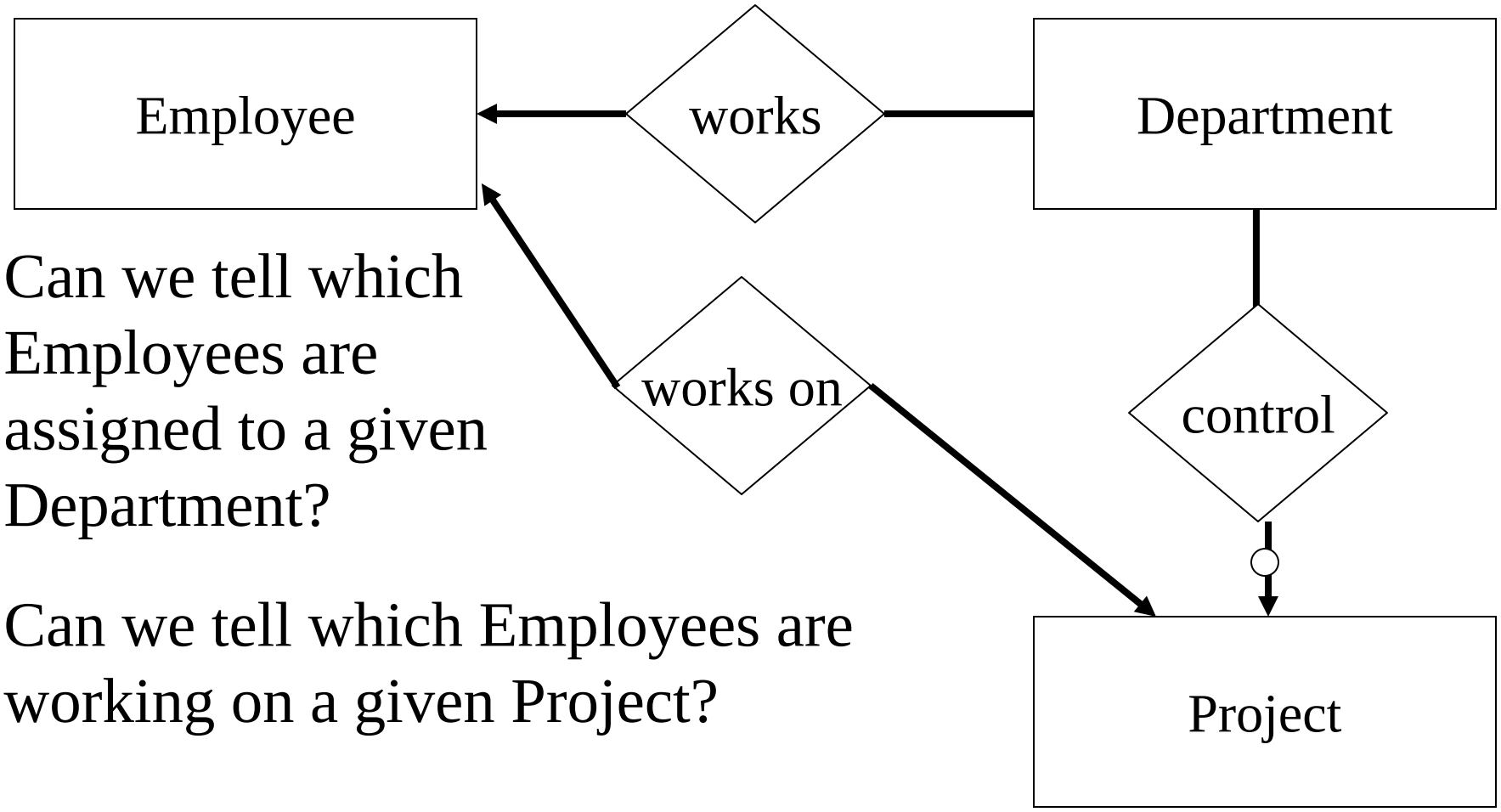
E-R Modelling (Alternate Notations)



E-R Modelling (Alternate Notations)



Using Relationship to Define Access Paths

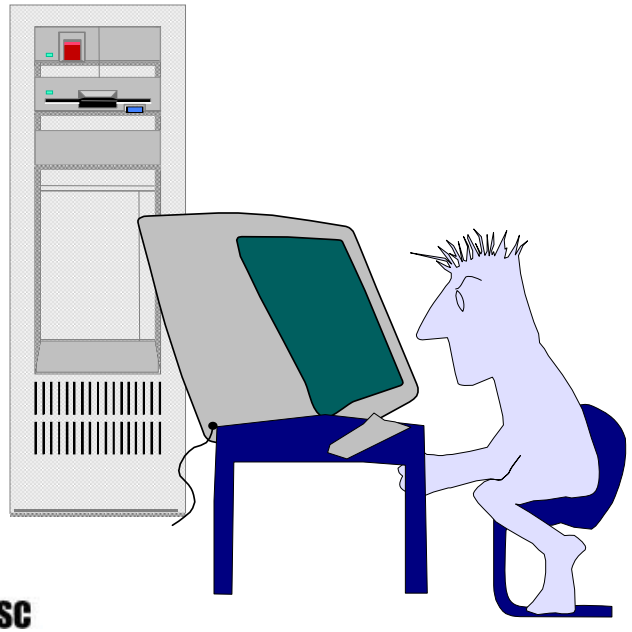


Can we tell which Employees are assigned to a given Department?

Can we tell which Employees are working on a given Project?

Can we tell which Employee work on Projects that do not belong to their Department?

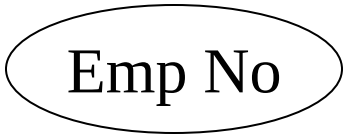
Detail Conceptual Data Model



Attributes

- Attribute
  - A property or characteristic of an entity type that is of interest to the organisation
- Simple Attribute
  - An attribute that cannot be broken down into smaller components

e.g. Emp No

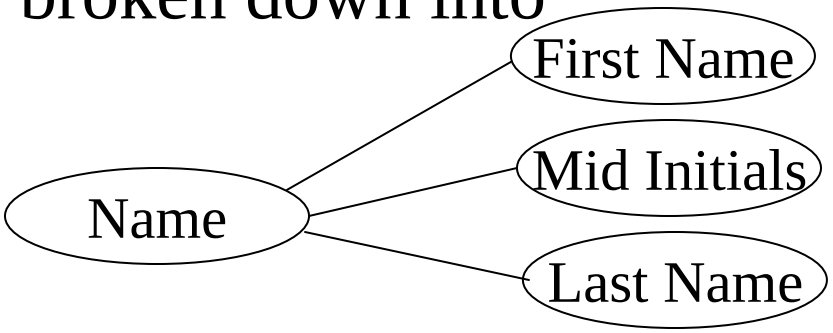


Attributes Cont'd

- Multi-valued Attribute
  - An attribute that may take on more than one value for a given entity instance

    e.g. Employee Skills, Qualifications

- Composite Attribute
  - An attribute that can be broken down into component parts



e.g. Address (Street, City, State, Postal Code)

Name (First Name, Middle Initials, Last Name)

Attributes Cont'd

- Stored Attribute
  - An attribute whose valued is stored in the database
- Derived Attribute
  - An attribute whose values can be calculated from related attribute values

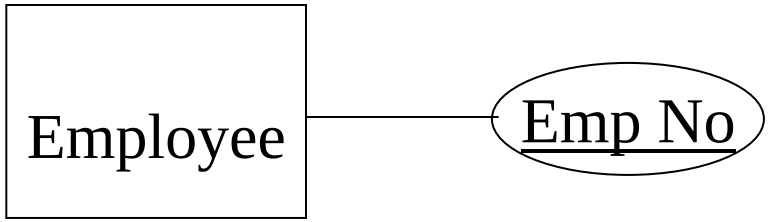
e.g. Years Employed (using Employed Date)

Age (using Date of Birth)

Identifier

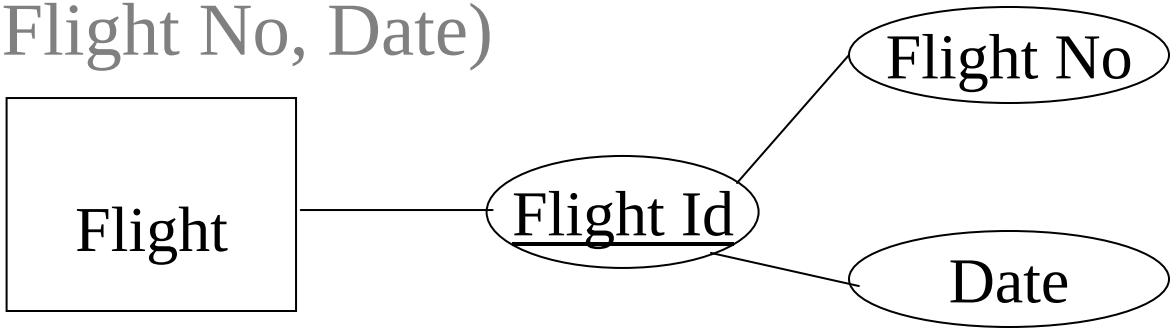
- Identifier
  - An attribute (or combination of attributes) that uniquely identifiers individual instances of an entity type

    e.g. Emp No



- Composite Identifier
  - An identifier that consists of a composite attribute

    e.g. Flight Id (Flight No, Date)



Identifier

- Choose an identifier that will not change its value over the life of each instance of the entity type
- Choose an identifier such that each instance of the entity type, the attribute is guaranteed to have valid values and not be null (or unknown)
- Avoid the use of so-called intelligent identifiers, whose structure indicates classifications, etc.
- Consider substituting single-attribute identifiers for large composite identifiers

Detailed Conceptual Design

Attributes

Department

Number
Location

Manager

Start date

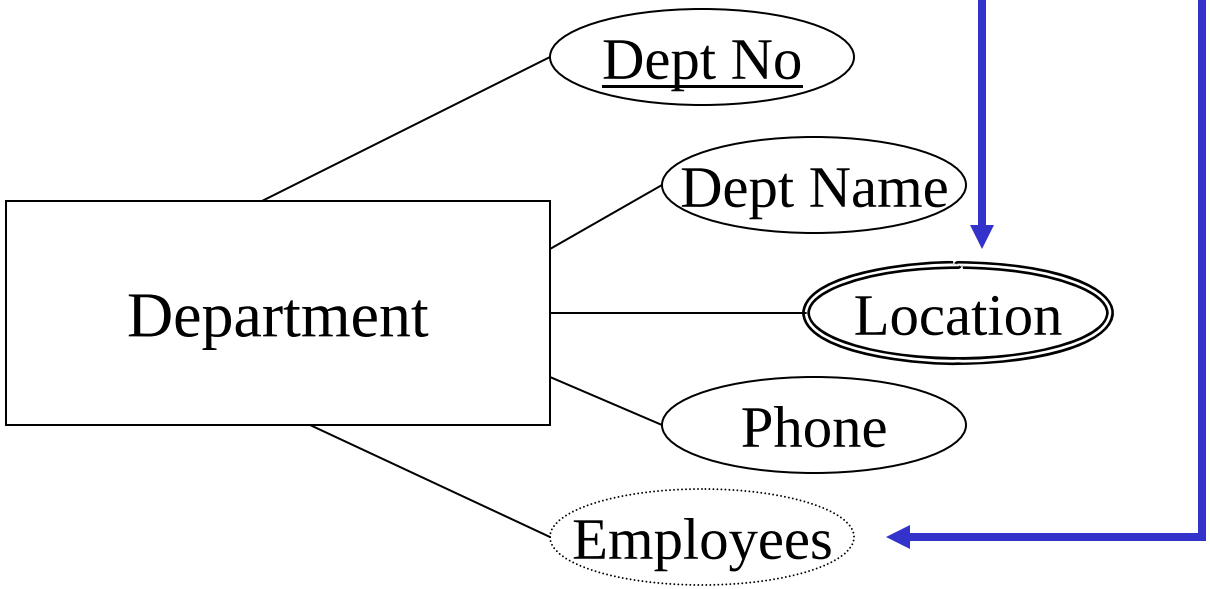
Name
Phone

Control Projects
Employees work for

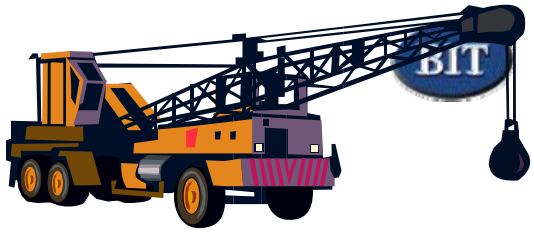
Number of Employees

Detailed Conceptual Design

Dept No	unique identifier of a dept.	Identifier
Dept Name	name of a department	Unique
Location	location of a department	Multi-valued
Phone	phone no. of a department	
Employees	no. of employees in a dept.	Derived



Detailed Conceptual Design



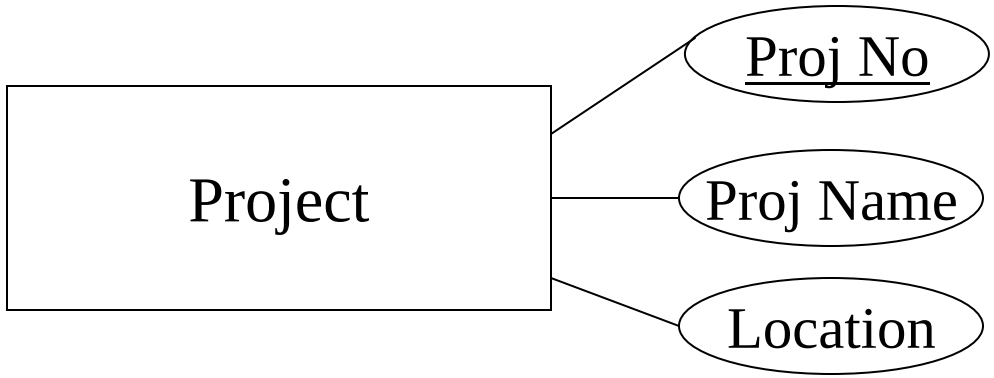
Project

Name
Number
Location

Department Control

Leader → *Employee*

Proj No	unique identifier of a project	Identifier
Proj Name	name of a project	Unique
Location	location of a project	



Detailed Conceptual Design

Employee

Name
National ID
Address
Salary
Sex
Birth Date

Works for Department
Supervise Employee

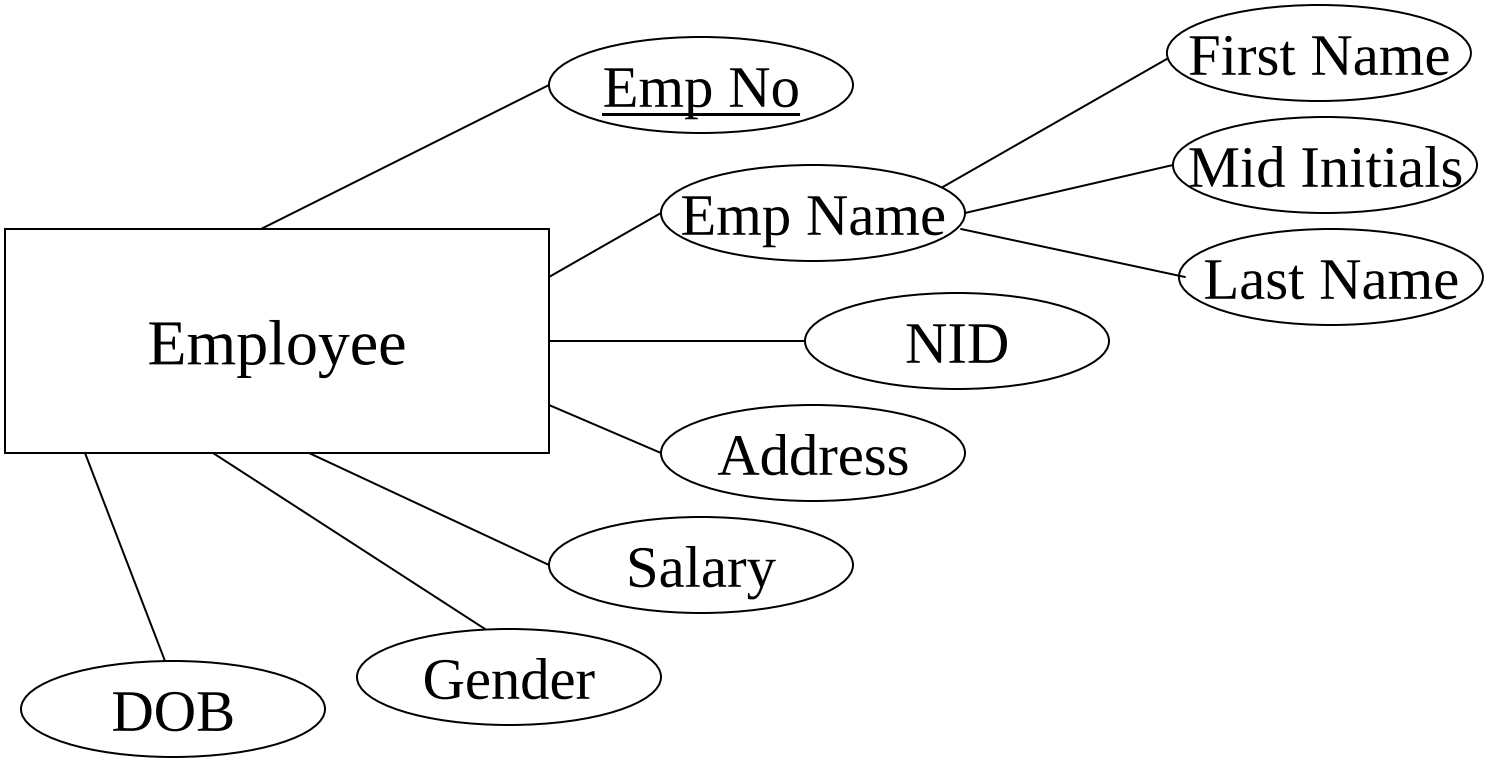
Emp No

Detailed Conceptual Design

Employee

Emp No	unique identifier of an emp.	Identifier
Emp Name	name of an employee	Composite
First Name	first name of an employee	
Mid Initials	middle initials of an employee	
Last Name	last name of an employee	
NID	national id of an employee	Unique
Address	address of an employee	
Salary	salary of an employee	
Gender	sex of an employee	
DOB	birth date of an employee	

Detailed Conceptual Design

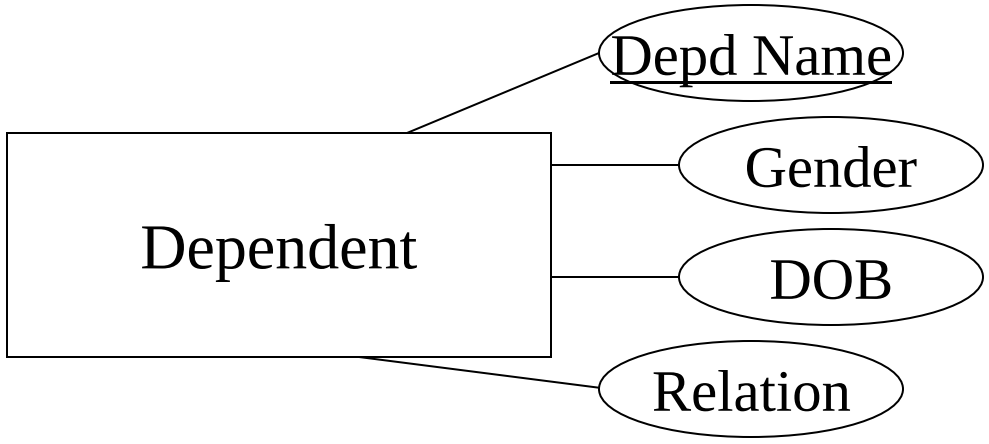


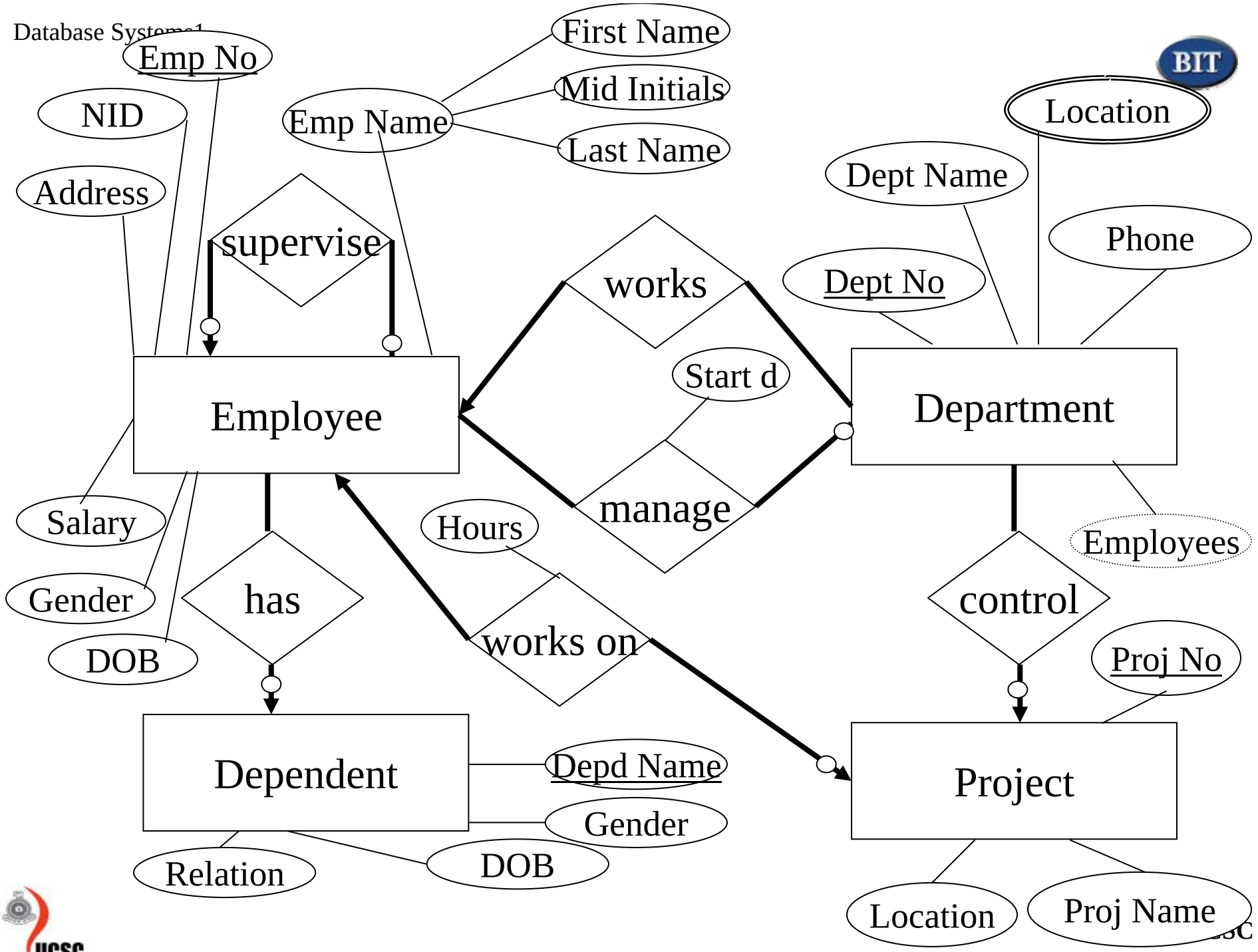
Detailed Conceptual Design

Dependent

Name	Sex	Birth Date	Relationship

Depd Name	name of a dependent	Part of Key
Gender	sex of a dependent	
DOB	birth date of a dependent	
Relation	relationship of a dependent to an employee	

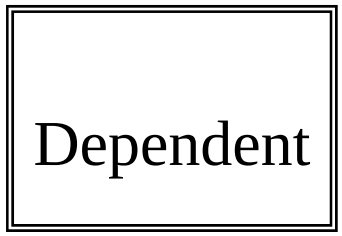



Emp No
First Name
Mid Initials
Last Name
Emp Name
NID
Address
supervise
works
Location
Dept Name
Phone
Dept No
Start d
Employee
Department
manage
Salary
Hours
Employees
Gender
has
control
works on
DOB
Proj No
Dependent
Depd Name
Gender
Project
Relation
DOB
Location
Proj Name

Entity Types

- Strong (Regular) Entity
  - An entity that exists independently of other entity types

- Weak Entity
  - An entity types whose existence depends on some other entity

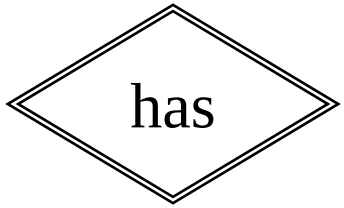


Entity Types

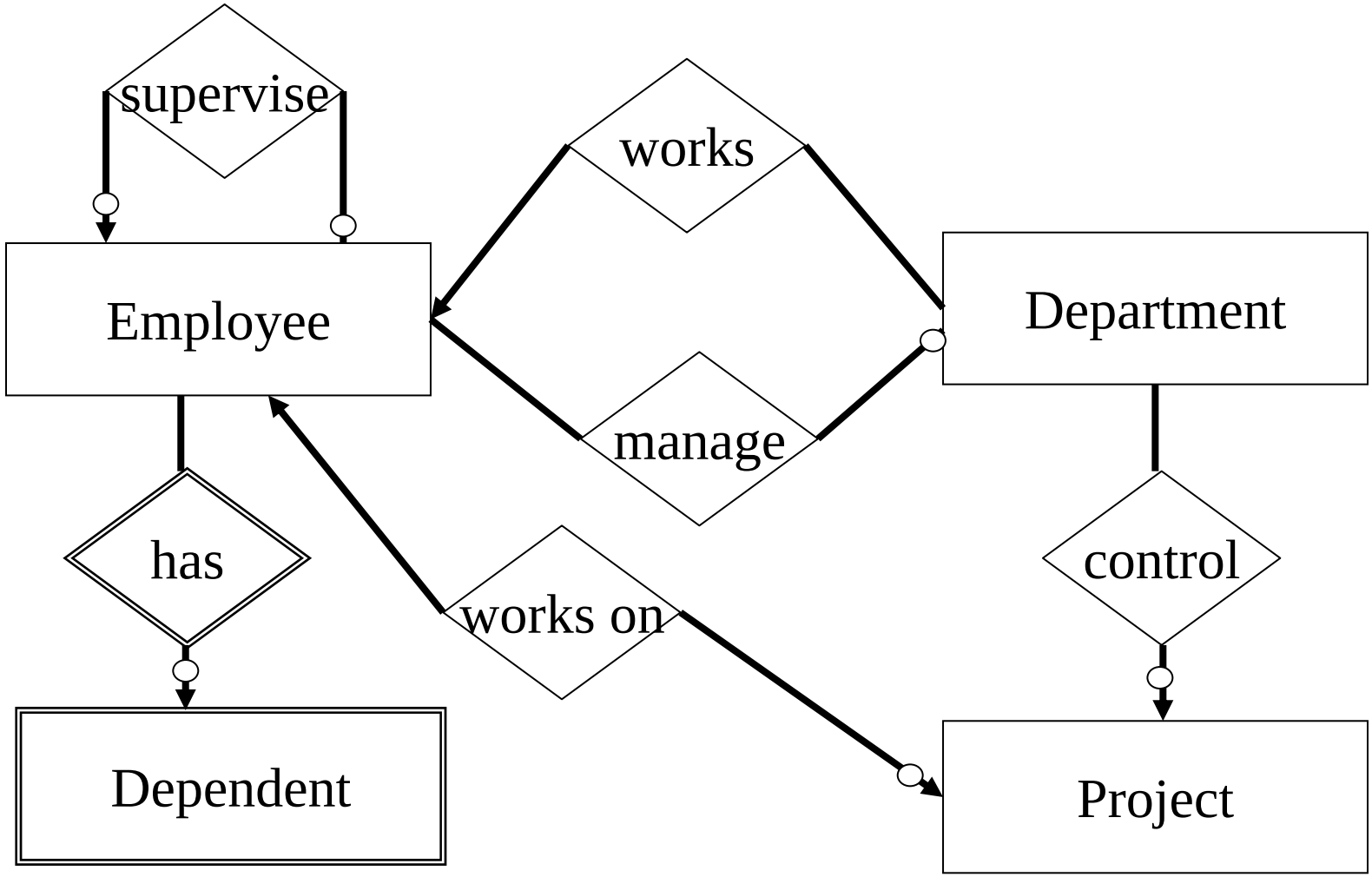
- Identifying Owner
  - The entity type on which the weak entity type depends

  e.g. Employee is the Owner of Dependent

- Identifying Relationship
  - A relationship between a weak entity type and its owner



Conceptual Design showing weak entities



Sample Entity Definitions

- Name: **Department**

  Type: Regular

  Definition: a department of an organisation

  Identifier: Dept_No

- Name: **Dependent**

  Type: Weak

  Definition: a person who is a dependent of an employee and entitle for insurance

  Identifier: Depd_Name (partial only)

Sample Attribute Definitions

- Name: **Emp_No**

  Domain: employee identities

  Definition: unique identifier of an employee

  Null: No

- Name: **Emp_Name**

  Components: First_Name, Mid_Initials, Last_Name

  Definition: a partial identifier of a name

  Null: No

Sample Relationship Definitions

- Name: **Works_for**

  Type: binary 1:M

  Definition: associates each employee with a department

  Constraint: each employee must be attached to a department

  Attributes: none

- Domain Constraints
  - A specification of the characteristics of the data values that can be associated with one or more attributes

Sample Domain Constraints

- Name: **Employee identities**

  Data Type: character

  Length: 5

  Allowable Characters: digits

Sample Domain Constraints Cont'd

- Name: **Last name**

  Data Type: character

  Max Length: 20
- Name: **DOB**

  Data Type: date

  Format: dd/mm/yyyy

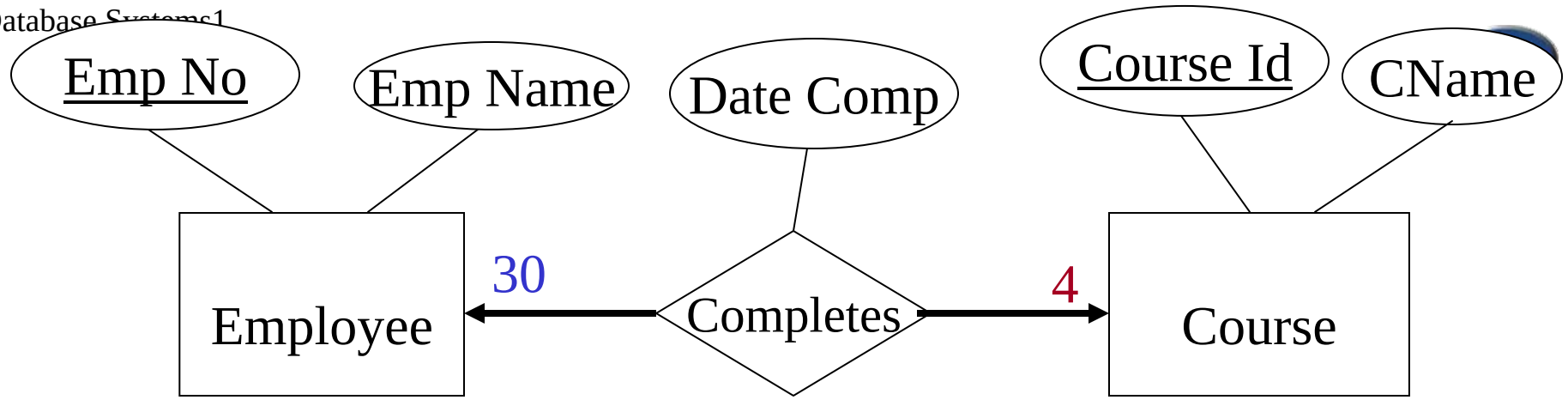
  dd = day

  mm = month

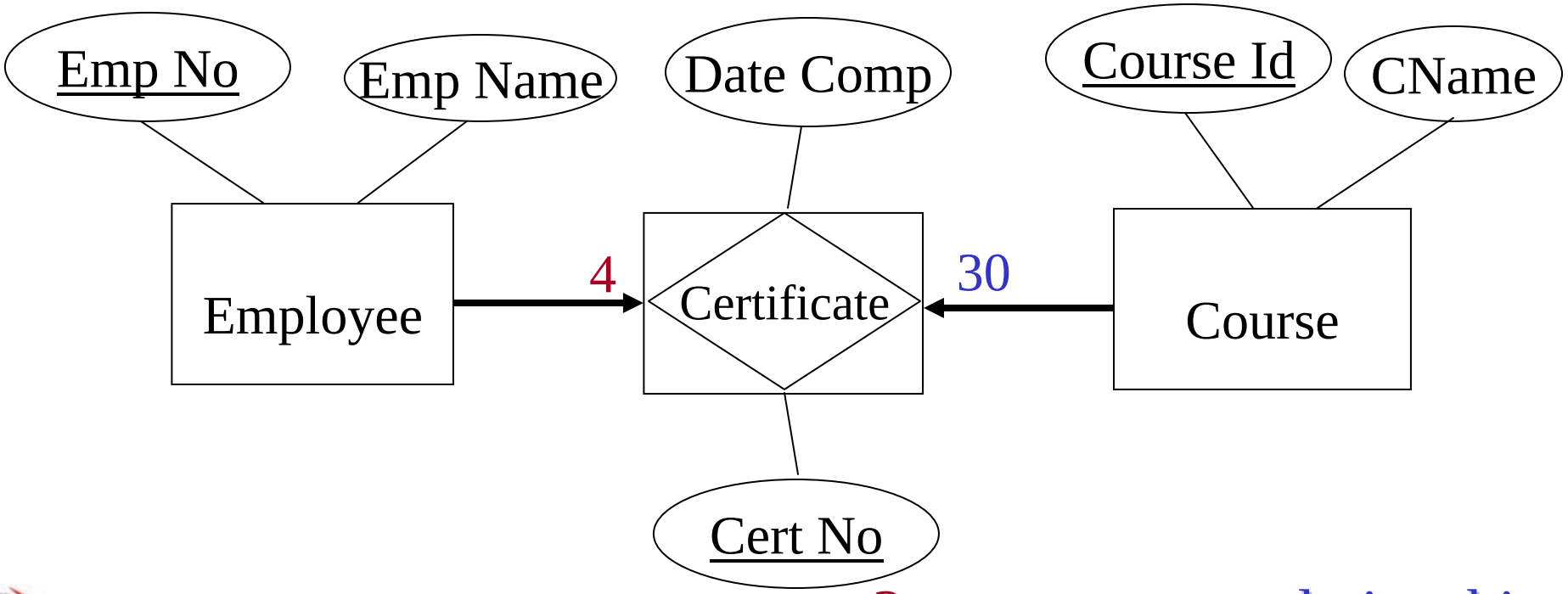
  yyyy = year

Associative Entity

- An entity type that associates the instances of one or more entity types and contains attributes that are peculiar to the relationship between those entity instances



1 many to may relationship



2 one to many relationships

Associative Entity

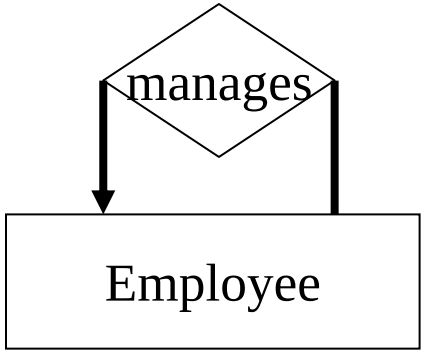
- All of the relationships for the participating entity types are “many” relationships
- The resulting associative entity type has independent meaning to end users, and preferably can be identified with a single-attribute identifier
- The associative entity has one or more attributes, in addition to the identifier
- The associative entity participates in one or more relationships independent of the entities related in the associated relationships

Relationships

- Unary Relationship
  - A relationship between the instances of a single entity type

  e.g. Person is married to a Person (1:1)

  Employee manages Employees (1:M)

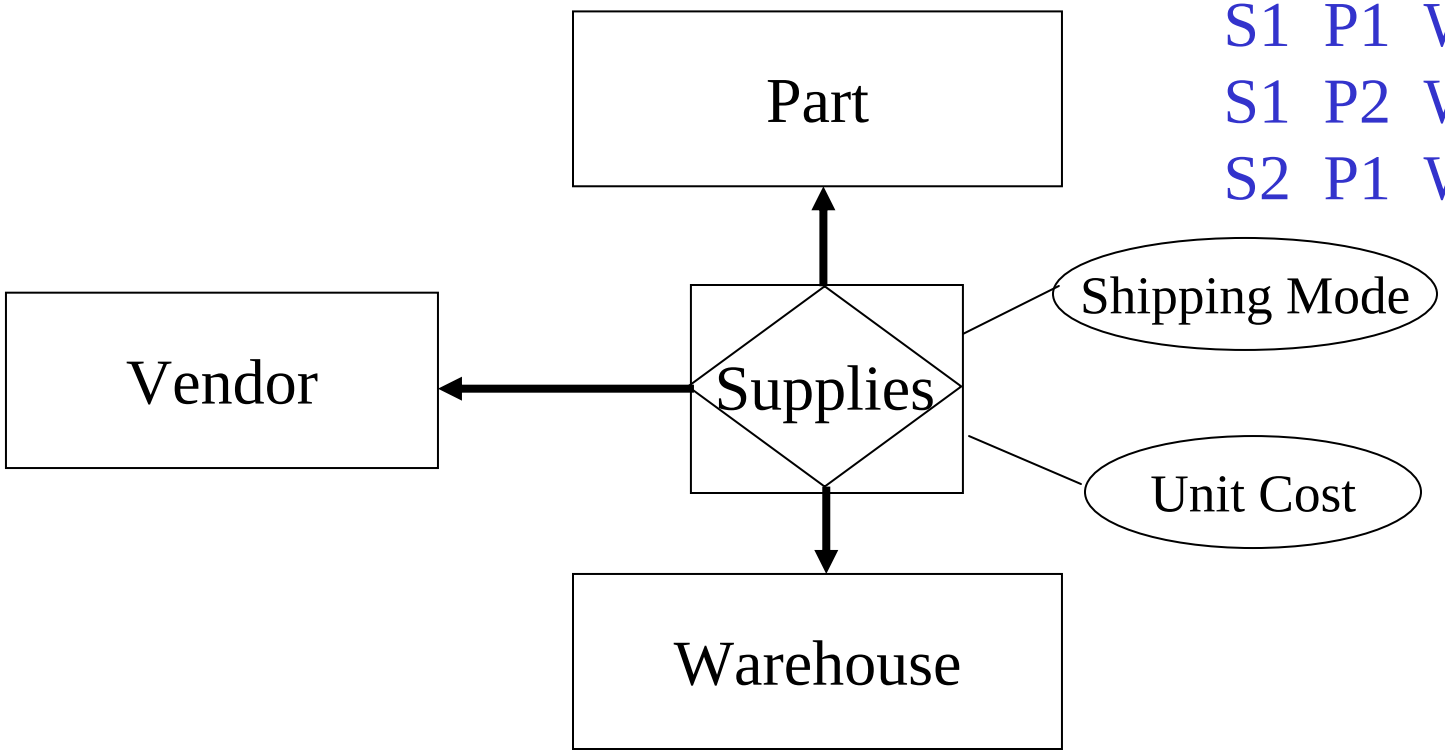


- Binary Relationship
  - A relationship between the instances of two entity types

Relationships

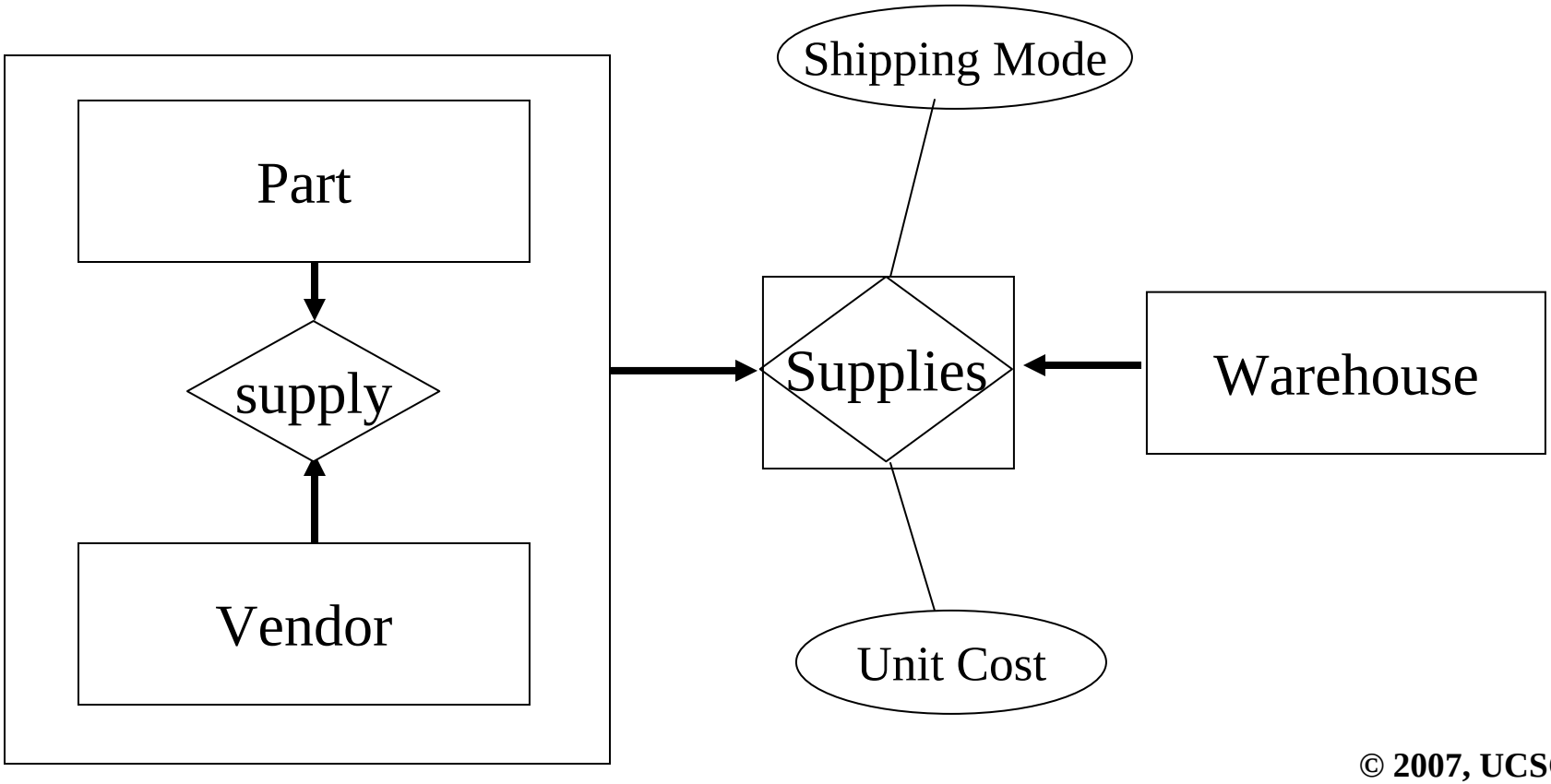
- Ternary Relationship
  - A simultaneous relationship among the instances of three entity types

S1 P1 W1 Land 10
S1 P1 W2 Sea 15
S1 P2 W1 Air 20
S2 P1 W1 Air 15



Relationships

- Ternary Relationship
  - can be treated as two many to many relationships



Enhanced ERM

- Enhanced Entity-Relationship Model
  - The model that has resulted from extending the original E-R model with new modelling constructs

Most important modelling construct incorporated is **Supertype / Subtype relationships**

Subtype

- A sub-grouping of the entities in an entity type that is meaningful to the organisation and that shares common attributes or relationships distinct from other sub-grouping. e.g. Student → Graduate, Undergraduate

Enhanced ERM Cont'd

Supertype

- A generic entity type that has a relationship with one or more subtypes. e.g. Student

Attribute Inheritance

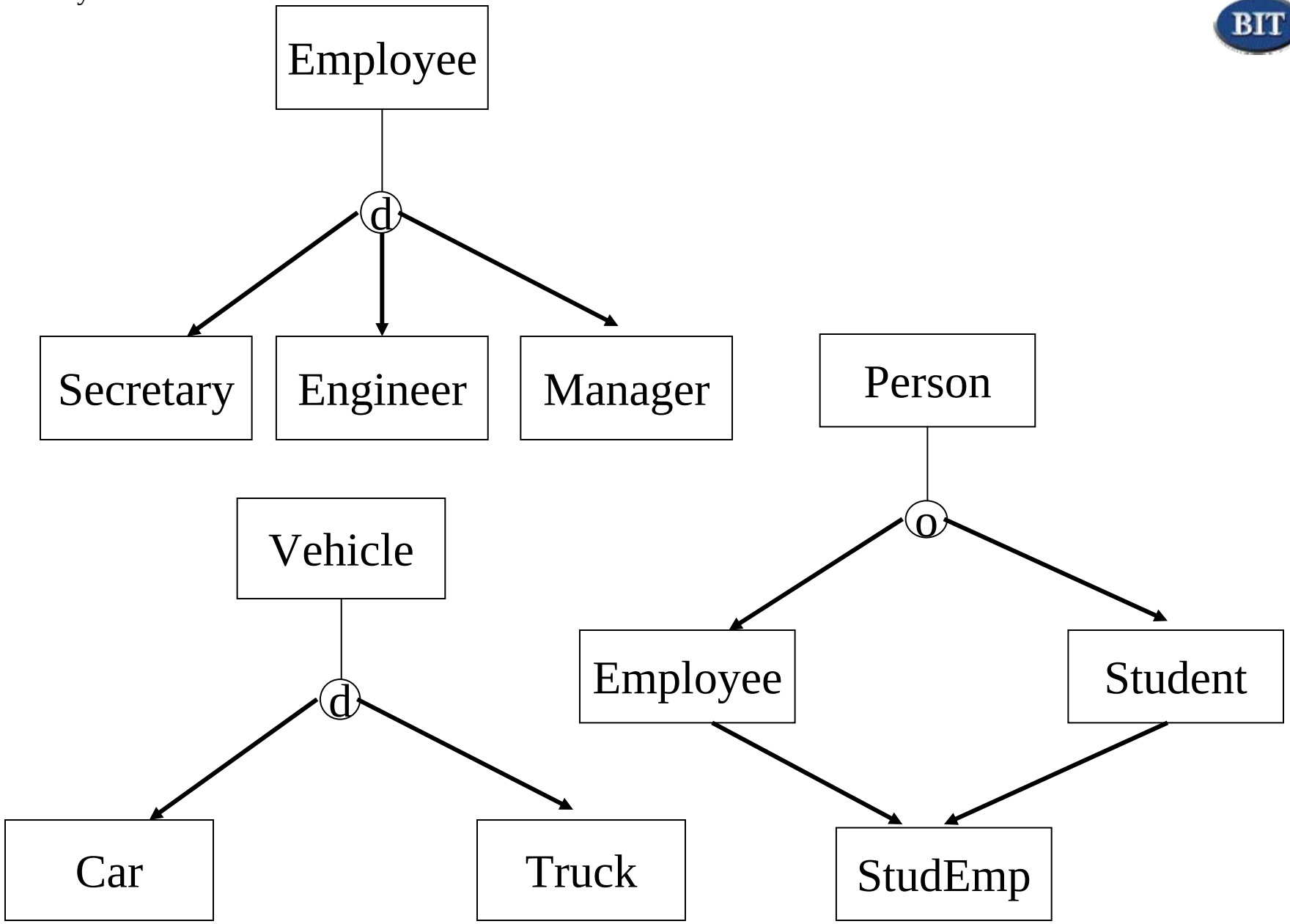
- A property that subtype entities inherit values of all attributes of the supertype

Generalisation

- The process of defining a more general entity type from a set of more specialised entity types

Specialisation

- The process of defining one or more subtypes of the supertype and forming supertype/subtype relationships


Employee
d
Secretary
Engineer
Manager
Person
o
Vehicle
d
Employee
Student
Car
Truck
StudEmp